国内图书分类号：TP301.6
国际图书分类号：681.14

密级：公开

西南交通大学
研究生学位论文

子空间聚类集成的关键技术研究

年　　级　二〇一一级
姓　　名　刘 波
申请学位级别　工程硕士
专　　业　计算机技术
指 导 老 师　王红军 副研究员

二零一四年五月

Classified Index: TP301.6
U.D.C: 681.14

Southwest Jiaotong University

Master Degree Thesis

RESEARCH ON KEY TECHNOLOGIES OF SUBSPACE CLUSTER ENSEMBLE

Grade: 2011

Candidate: Liu Bo

Academic Degree Applied for: Master Degree

Speciality: Computer technology

Supervisor: Wang Hongjun

May, 2014

西南交通大学

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权西南交通大学可以将本论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复印手段保存和汇编本学位论文。

本学位论文属于

1．保密□，在年解密后适用本授权书；

2．不保密☑，使用本授权书。

（请在以上方框内打"√"）

学位论文作者签名：刘波　　指导老师签名：

日期：2014.5.21　　日期：2014.5.21

西南交通大学硕士学位论文主要工作（贡献）声明

本人在学位论文中所做的主要工作或贡献如下：

(1) 本文提出最小冗余特征子集的子空间划分法，通过减少实验中数据在聚类过程中的冗余度来改善聚类结果。

(2) 本文提出基于属性最大间隔的子空间划分法，通过计算数据属性之间的互信息，利用标准化后的互信息建立特征矩阵，最后用最大间隔法来实现对数据集的子空间划分，并以此提高聚类精度。

(3) 本文通过对比研究已有的集成模型，筛选出在子空间聚类集成中性能最佳的集成模型对基聚类成员进行集成操作，以此得到最佳的子空间聚类集成结果。

本人郑重声明：所呈交的学位论文，是在导师指导下独立进行研究工作所得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中作了明确说明。本人完全了解违反上述声明所引起的一切法律责任将由本人承担。

学位论文作者签名：刘波

日期：2014.5.21

摘 要

子空间聚类算法能有效减少数据冗余和不相关属性对聚类过程的干扰，从而提高在高维数据集上的聚类效果。已有的子空间聚类算法主要强调在各个子空间中簇的发现，通常忽略了子空间的划分。高维数据中，子空间划分的正确与否，直接影响到高维数据聚类正确率的高低，因此想要提升高维数据的聚类准确率，就必须采用正确的子空间划分方法来对高维数据进行子空间划分。

本论文提出了两种划分数据子空间的方法，第一种是基于最小冗余特征子集的子空间划分法，第二种是基于属性最大间隔的子空间划分法。基于最小冗余特征子集的子空间划分法是在 K-means 算法的基础上改进的，将计算数据特征变量间的互信息替换 K-means 算法中计算数据特征变量间的距离，根据数据特征变量间互信息值的大小来对数据进行子空间划分，用这种方法划分出来的子空间叫做最小冗余特征子集。

基于属性最大间隔的子空间划分法是通过计算数据两两属性间的互信息，然后将属性间的互信息值归一化后构建一个特征矩阵。得到特征矩阵后，再利用网格划分法将特征矩阵划分成不同的子块，通过搜索子块中互信息的最大值得到数据集中两个属性变量之间的最大信息系数，最大信息系数体现了两个属性之间关联性的大小，关联性越大，属性间间隔越小，关联性越小，属性间间隔越大，因此在得到最大信息系数后，我们就可以利用最大间隔原理来对数据集进行子空间划分。

最后，通过实验验证本论文提出的两种子空间划分方法的有效性，采用 UCI 和 NIPS2003 比赛等数据来进行实验，实验结果表明，在大多数数据上采用基于最小冗余特征子集法和属性最大间隔法对数据集进行子空间划分后得到比其他子空间聚类算法更好的聚类结果。

关键词：聚类；子空间聚类；最小冗余特征子集；属性最大间隔

Abstract

Subspace clustering algorithm can reduce the influence of redundancy and irrelevant attributes effectively during the clustering process, and improve the clustering accuracy. Existing subspace clustering algorithms emphasize the find of clusters in all subspace, and ignore the divide of subspace. In a high-dimensional dataset, the divide of subspace affects the clustering accuracy of high-dimensional dataset directly. In order to improve the clustering accuracy of high-dimensional dataset, a correct subspace dividing method must be used to divide the subspace.

In this thesis, two methods are proposed to divide dataset. The first is the method of subspace dividing based on minimum redundancy feature subset, and the second method based on maximum margin. The method based on the minimum redundancy feature subset was improved based on K-means algorithm. We calculate the mutual information between data characteristic variables instead of calculating the distance between characteristic variables, according to the value of mutual information, data subspace is divided, and subspace divided by this method is called the minimum redundancy feature subspace.

The method of subspace dividing based on the maximum margin is determined by the mutual information between each pair of attributes. Characteristic matrix is built based on the mutual information value between each pair of attributes. Meshing method is used on characteristic matrix to get different sub-blocks. Maximal information coefficient is obtained through research the maximal mutual information value of these sub-blocks. Maximal information coefficient reflects the correlation between each pair of attributes. The greater relevance, the smaller margin; The smaller relevance, the greater margin. According to the maximal information coefficient, the subspace could be divided based on the maximum margin principle.

Finally, experiments are performed to verify the validity of two subspace dividing methods. Experiments on UCI and NIPS2013competition datasets show that the method of subspace dividing based on minimum redundancy feature subset and the method of subspace dividing based on maximum margin on most datasets perform better than other subspace clustering algorithms.

Key words: Clustering; Subspace Clustering; Minimum Redundancy Feature Subset; Attribute Maximum Margin

目 录

第 1 章 绪论

1.1 研究的背景和意义

随着社会经济的快速发展，社会活动中如：web数据，物联网数据，证券市场分析、通讯数据管理等产生的数据量越来越大，如何从这些海量的数据中找出有用的信息来指导我们今后的发展，成为几乎所有领域的共同需求，正是在这样的大趋势下，机器学习的作用日渐重要，受到了研究人员广泛的关注。

聚类是机器学习中的一项关键技术，它被广泛用来探索数据内部结构。聚类的目的是将一个数据集划分为多个簇，使得同一簇内的数据对象之间具有较高的相似度，而属于不同簇的对象之间相似度低。其中，通常采用间隔距离的大小来度量两个数据对象之间的相似度[1]。目前使用比较广泛的聚类方法有 K-means[2]，K-medoids[3]，FCM(Fuzzy C-Means)[4]等。现阶段聚类分析被广泛地运用在统计学、模式识别、机器学习等领域中[5]，信息技术的发展使得数据采集和存储变得更加快捷和简单，也由此产生了大量维数多、规模大的复杂数据集。例如，文本挖掘中由 VSM（向量空间模型）[6]表示的文档向量可能具有几百甚至上千个特征，因此高维数据集中往往存在数据稀疏性以及“维数灾难”[7 8]等问题。此外，不同类别的数据样本经常与不同的属性子集（子空间）相关，因此，在数据冗余和不相关属性的影响下，传统的聚类算法无法有效解决高维数据集上的聚类问题[9]，高维数据聚类问题成为目前研究人员面临的重要挑战任务之一。

高维数据空间中经常存在许多不相关的属性，使得要搜索的簇类往往隐藏在一些低维子空间中，且不同的簇类其关联的子空间往往也不相同[8 10]。在高维空间挖掘隐藏在不同低维子空间中簇类的过程，称为子空间聚类。子空间聚类算法能有效减少数据冗余和不相关属性对聚类过程的干扰，从而提高在高维数据集上的聚类结果。在近十多年的时间里，国内外学者提出了许多关于子空间聚类的方法，如基于网格划分、数据密度划分等方法。子空间聚类算法使得高维数据的处理效果得到了很大的提高，因此子空间聚类在近十多年的时间里得到了快速的发展，成为了数据挖掘领域一个新的研究热点。

尽管目前提出的子空间聚类及其改进算法对高维数据的处理能力有了很大的提升，但在现实中还没有一个单一的算法能够对任意数据结构进行十分有效的处理。如有些子空间聚类算法对数据点密度比较高的数据处理的结果比较好，有些算法对数据点均匀分布在各维的数据处理能力比较强。如果聚类时采用的聚类算法不适合某些具有特殊分布的数据集，就会导致得到较差甚至错误的聚类结果。在近年中，因为受到日益成熟的集成技术的启发，有研究者提出了聚类集成这个概念。聚类集成算法综合

几种不同聚类算法或者同一种算法通过多次运算产生的不同基聚类结果，因此它获得聚类结果比单一的使用一次聚类算法得到的聚类结果更好。大量的实验结果证明，聚类集成算法能够体现更好的鲁棒性、稳定性、可扩展性[11]。

本论文依托国家自然科学青年基金项目“半监督聚类集成的关键技术研究(61003142)”、国家自然科学基金地区基金项目“藏文 Web 信息的社会网络动态演化机理研究（61262058）”、国家自然科学基金面上项目“基于半监督学习的聚类集成机理及高效算法研究 (61170111)”、西南交通大学牵引动力国家重点实验室自主研究课题“基于云计算的海量高铁数据处理关键技术研究”四个课题，是其中重要的组成部分。如何设计合理有效的子空间划分法，选取合适的聚类集成模型综合各聚类算法的优点，在很大程度上影响着子空间聚类结果的质量。因此子空间的划分、聚类集成模型的选取是现阶段子空间聚类集成研究的一个重点问题，也是本论文研究的意义所在。

1.2 国内外研究现状

1.2.1 子空间聚类研究现状

子空间聚类的概念是 R.Agrawal 在 1998 年首次提出。子空间聚类是在高维数据集中搜索其所有低维子空间中的数据点集合，其定义如下：

定义 1：高维空间 D 中，在其子空间 ς 中数据点密度较大的点集称为类，记作 C。

令 $C_1, C_2..., C_K$ 为空间 D 中的 K 个子空间聚类，$i=1,...,K$,则由类的定义，我们可知类中的点应该具有以下两个性质:

性质 1：类中点的密度大于类外点的密度。

$$Density(p,\varepsilon) > Density(q,\varepsilon) \quad p \in C_i, q \in noise \tag{1-1}$$

性质 2：同类中点的密度差异小于类外点的密度。

$$\left|Density(v,\varepsilon) - Density(p,\varepsilon)\right| < \left|Density(v,\varepsilon) - Density(q,\varepsilon)\right| \quad v, p \in C_i, q \in noise \tag{1-2}$$

为了更加形象地展示子空间聚类问题，我们用图 1-1 来进行说明。

图 1-1 表示数据点在一个二维空间中的分布情况。这个二维空间分别投影到 x 轴和 y 轴上，形成 $\{x\}$ 和 $\{y\}$ 两个子空间，在这两个子空间中，分别搜索到 C_b, C_c, C_d 和 C_a 四个类。结合 $\{x\}$ 和 $\{y\}$ 两个子空间可知 C_c 投影到 x 轴上时数据点比较分散，所以在二维空间中 C_c 不能被定义为一个类。因此在这个二维空间中，只有 C_{ab}, C_{ad} 可以被视为子空间聚类。

在子空间概念提出之前，已有的聚类算法根据算法思想的不同，可以大致归为五个大类[12]，这五类算法分别是：迭代法、层次法、基于密度法、基于网格法、基于模

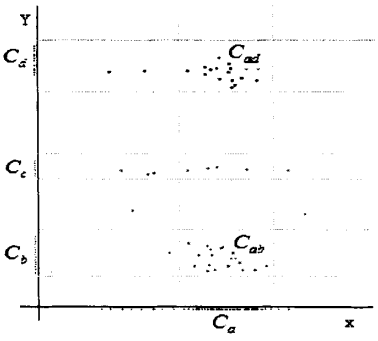


图 1-1 二维空间中的子空间聚类

型法。近年，随着大数据，复杂形状和类型以及混合数据的出现，对于聚类算法在这些数据上进行聚类的伸缩性、有效性以及高维聚类分析技术的研究得到研究人员的广泛关注。其中，高维数据的研究是难点，也关系到聚类算法能否在很多领域得到应用。对高维数据进行聚类时，有两个问题在利用传统聚类算法时会经常遇到，分别是：（1）高维数据集中通常存在大量相互之间不相关的属性，这些不相关的属性就使得数据中的簇不可能存在所有的数据维之中；（2）传统聚类算法对数据进行聚类时主要是利用数据点间的距离相似度来作为聚类标准，但是在高维数据空间中，数据点的分布较低维空间的数据点分布更稀疏，这样就使得数据点间的空间距离几乎相等的，这样就使得在高维数据空间中无法基于数据点间的距离来构建簇。目前一般使用两种方法解决以上两个问题：特征转换和特征选择[13]。特征转换法通过将原数据的维利用线性合并的方法进行合并至 M 个新维，从而达到减少维的目的。这一类方法主要包括主成分分析和奇异值分解等策略。特征转换算法主要有三个缺点：一是数据合并的维数 M 比较难确定；二是高维空间中许多簇会被空间中大量不相关的维所遮盖，数据中大量不相关维的干扰使得对数据进行聚类非常困难；三是在聚类的过程中产生许多没有实际意义的簇，这些没有实际代表意义的簇就会影响最终的聚类结果质量。因此如果想要利用特征转换算法得到较好的聚类结果，就需要在已知数据的多数维都是比较相关的高维数据中进行，只有在这样的条件下利用特征转换算法才能得到比较理想的聚类结果。特征选择与特征转换的区别点在于特征选择只在数据中相关的子空间中挖掘数据之间的关系，因此利用特征选择算法对数据进行降维操作比利用特征转换算法更有效地减少数据的维数。特征选择算法利用一些评价标准来评价其在数据空间中搜索到的不同特征子空间，通过评价结果来发现聚类所需要的簇。

最早的子空间聚类算法是由 R.Agrawal 等人提出的 CLIQUE[14]，该算法是基于数据划分网格、数据密度的思想。CLIQUE 算法通过查找参数值大于给定阈值的网格，然后再通过合并满足条件的网格来产生子空间，并按照将这些数据子空间中数据点的多少进行顺序排列，再使用 MDL（Minimum Description Length）准则将不满足需求

的子空间剪切掉。CLIQUE 算法找到的子空间可以用一个 DNF（Disjunctive Normal Form）式子表示，而且不必在聚类之前清楚子空间的数目。ENCLUS[15]、MAFIA[16 17 18]，是 CLIQUE 的改进算法。ENCLUS 算法根据子空间的熵值来进行能否形成子空间判断，通常情况下，可以形成簇的子空间其熵值都会低于无法形成簇的子空间的熵值。MAFIA 算法是通过利用数据在空间中的分布情况来动态地调整判定数据子空间网格的大小，利用动态调整网格的方法不仅可以提高聚类算法的效率，还能提高聚类结果的质量。此外通过引入数据并行化处理方式，使得 MAFIA 算法具有较强的伸缩性，在与 CLIQUE 算法相比较时，MAFIA 算法的执行速率有较大的优势。CBF[19]算法通过在一个基于过滤的索引结构中记录下箱子的有效信息来增强聚类算法的索引性能，因此在 CBF 算法中需要输入的参数有两个，分别是箱阈值和项阈值。项阈值决定算法的检索时间，因为项阈值是确定数据每一维中箱子的数量，所以每一维中箱子的数量分的越多，算法所用检索时间就越少，相反，每一维中箱子所分数量越少，算法所用的检索时间就越长；箱阈值确定的是每一个箱子中的最少数据点数量，只有当箱子中数据点数量大于箱阈值的箱子才会被当作簇的一部分。CBF 算法与 CLIQUE 算法相比较，CLIQUE 算法在精度方面会稍高，但是 CBF 算法在检索速度方面却比 CLIQUE 算法要快很多。CLTree[20]算法是通过构造决策树来判定数据空间里面密集的子空间和稀疏的数据单元，CLTree 算法与 CBF 算法有相似之处在于两种算法都需要对数据的每一个维进行箱子数量的划分，只是 CLTree 算法是通过构造决策树的方法来对数据集的每一维进行箱子的划分，并利用这些划分出来的箱子中数据点密度最小的区域作为簇的分割线，这样就能够在簇的划分过程中尽可能小的减少数据点的丢失。DOC[21]算法同时采用了自顶向下和自底向上的搜索方法，DOC 算法在网格策略中采用的是自底向上的方法，而在改善簇质量方面采用的是自顶向下方法。两种不同搜索方法的结合，能够使得算法的聚类效果得到更大的改善。DOC 算法利用投影的方式将数据分成不同的投影簇，将每一个投影簇看成是一个簇对（C，D）的形式，C 代表的是数据的一个子集，D 代表的是数据子集 C 的相关维的子集，DOC 算法的目的就是寻找到在集合 D 中数据点子集 C 具有很强凝聚性的簇对。PROCLUS[22]算法采用自顶向下搜索方法，从数据样本中选择 K 中心点，通过不断的迭代和改进来提升簇的质量。因此在 PROCLUS 算法执行过程中需要以下三个步骤：初始化阶段、迭代阶段、改进阶段。与 CLTree 算法和 DOC 算法适合超矩形形状数据不同的是，PROCLUS 算法更加适合查找数据分布为超球面形状的簇，这样的簇表示为某一点为中心，某一长度为半径的所有子空间的数据点集合。因为 PROCLUS 算法以寻找到簇的维数的平均数量作为参数，所以要求寻找到的这些超球面形状的簇的大小一定尽可能的相似。ORCLUS[23]算法是 PROCLUS 算法的改进，用于寻找非轴平行的子空间。ORCLUS 算法可以分为三步：分配簇、确定子空间、子空间合并。运用该算法时，必需事先知

道聚类的数目和数据的维度数。现阶段聚类算法在聚类过程中使用的都是欧式距离或者曼哈顿距离来衡量数据对象之间的距离，但是 FINDIT[24]算法并未使用这两种常用的距离衡量标准，FINDIT 算法使用的是 DOD（Dimension oriented distance）衡量标准，利用 DOD 标准度量的好处在于它不仅可以表示出数据对象值之间整体的差别，还可以体现出数据对象间维度的差别。与 ORCLUS 算法相同， FINDIT 算法也是分为三个阶段来执行的，分别是抽样阶段、簇形成阶段、最终簇形成阶段。δ-Clusters 算法[25]与现阶段很多的聚类算法在对数据对象间相似性的判断上有明显的不同，δ-Clusters 算法搜索簇是根据子空间里数据点的属性是否一致来作为评判标准的。该算法开始把子空间划分为多个初始簇，然后重复调用簇改善过程，该过程通过彼此之间交换对象或属性来增加各个簇的质量。每一步得到的最好的簇成为下一步的起点，直到所有簇没有进一步改进，则该过程结束。COSA[26]算法是为每一个数据点的每一维赋权重，从具有相同权重值的维度开始，该算法检索每一个点邻近的 K 个点，每一个数据点的邻近点被用来计算该点的对应维的权重值。在 KNN 簇中离散度较小的维度，将被赋予更高的权重值，且这些权重值将被用于计算在 KNN 簇中更新数据点间距离的成对点的维度权重值。这是一个利用数据点间产生的新距离不断重复的过程，一直到最后权重值稳定为止。K-Subspace Clustering[27]是一个利用 K-means 聚类方法去处理线性、平面、球形簇的子空间聚类方法。该方法延续了 K-means 聚类的特点：容易实现和快速收敛。该方法能够用于聚类全部球形簇和不同文献中绝大部分形状的簇。Non-Redundant 算法[28]是用于识别无冗余且相关的子空间，利用这个方法可以避免产生重复的聚类结果，使最终的聚类结果更加准确。Ehsan Elhamifar 等人在文献[29]中提出了 SSC 算法，该算法是基于每一个数据样本点在数据联合子空间中都可以将数据进行稀疏表示这一事实，在将数据进行稀疏表示的基础上对从高维空间中搜索到的一系列线性或者仿射子空间进行聚类。Mahdi Soltanolkotabi 等人在 SSC 算法的基础上提出了 RSC 算法[30]。RSC 算法是一种针对噪声数据进行聚类的子空间聚类的算法，并且利用新的理论原理来证明方法的可靠性，这种新的理论原理是利用几何功能分析方法来对聚类结果进行分析评价，实验结果表明 RSC 算法能够准确地还原出隐藏着的数据子空间，且这些子空间均满足算法关于子空间方向或者每一个子空间包含的数据样本点数量等最低要求。受到 SSC 算法的启发，Wang 等人在文献[31]中提出了 NSSC 算法。在数据的联合子空间未带标签的数据中加入噪声数据，然后利用 NSSC 算法对其进行处理，实验结果表明 NSSC 算法能够有效地搜寻出隐藏在噪声数据中的数据子空间。

1.2.2 子空间聚类集成研究现状

子空间聚类集成的提出是为了结合不同子空间聚类算法的优点产生较好的聚类

结果。2009，F.Gullo 等[32]提出了第一个子空间聚类集成算法 PCE，该算法主要目的是从投影聚类算法中衍生出一个合适的共识投影分区。通过把 PCE 转化为一个优化问题，这样可以不依赖任何特定聚类集成算法，还可以处理数据的软、硬聚类，以及数据不同特征的权重问题。2010 年，F.Gullo 等[33]提出了 PCE 的改进算法，其思想是通过在算法采用更有效的公式在保证效率优势的前提下减少单目标 PCE 与双目标 PCE 间正确率的差距。2011 年，F.Gullo 等[34]提出一个 PCE 的可替代公式，这个公式可以将基于对象和特征表示的簇结合起来，实质上是将这些簇与一个投影聚类方法和一个给定的集合之间的距离计算相结合。

1.3 本文主要研究内容和结构安排

本文主要对子空间聚类集成中数据的子空间划分方法进行研究，针对现阶段子空间聚类算法中存在的一些问题，本论文提出了两种子空间划分方法，并利用新方法得到的子空间聚类结果与已有的子空间聚类算法的聚类结果进行对比。全文一共由五部分组成，具体安排如下：

第 1 章描述了本论文的研究意义和背景，详细介绍了子空间聚类、子空间聚类集成在国内外的研究现状，最后给出论文的结构安排。

第 2 章主要介绍子空间聚类集成的相关理论研究，包括聚类、聚类集成到现阶段为止的研究现状，并重点分析了聚类集成中基聚类成员的产生以及共识函数的设计等问题。

第 3 章提出基于最小冗余特征子集的聚类集成，该方法利用最小冗余法来对数据集进行子空间划分，以此来减少数据在聚类过程中因为数据冗余对聚类结果的影响。

第 4 章提出基于属性最大间隔的子空间聚类，该方法通过计算数据属性间的互信息值来对数据进行子空间划分，该方法可以避免以往算法对一些分布较散的重要属性的错误划分，能最大限度的减少有效信息的丢失，从而保持数据的原有特性。

第 5 章测试了第 3 章和第 4 章提出算法的有效性。首先对比了不同共识函数对聚类集成结果的影响，在选择出实验所用共识函数后，进行了论文的验证实验，并在论文最后对实验结果进行了简要分析。

第 2 章 相关理论基础概述

2.1 聚类算法分析

2.1.1 聚类的概念与研究分析

聚类是将数据对象划分为不同类的过程，且划分类的特点是：同一类中的对象相似度最大，而类与类之间对象的相似度最小。类内的相似性越大，类间的差异性越大，划分类的效果就越好。

聚类算法作为数据挖掘领域里面的一个重要分支，其在医学图像，人工智能等领域有着广泛的应用。聚类算法与分类算法的不同之处在于分类算法是非常有效的识别数据对象的方法，但它需要人们花费大量的时间去搜集和标记训练对象，而聚类算法则是一种无监督学习行为，在运行聚类算法前，我们对需要进行聚类操作的数据集的内部结构是不知道的，需要划分的类也是未知的，聚类算法会根据数据对象之间相似性来自主进行类别的划分，不需要先验知识，这就更能方便快捷地处理现代社会产生的“海量”数据。

假设数据集 X 由一个 $N\times M$ 的矩阵表示，则 $X=\{X_1,X_2,...,X_N\}$ 表示数据集 X 中含有 N 个数据样本，$X=\{X_{.1},X_{.2},...,X_{.M}\}$ 表示数据集 X 包含有 M 维属性（特征）。任意选择一种聚类算法 φ 对数据集 X 进行聚类，可以将 N 个数据样本划分为 K 类，则类集合 C 可以表示为 $C=\{c_1,c_2,...c_k\}, c_i\neq\varnothing\ (i=1,2,...k)$。因为是将数据集划分为 K 类，且每一个数据样本均只属于一个类中，则集合 C 及其子集合 C_i 满足：$X=C_1\cup C_2\cup...\cup C_k$ 且 $C_i\cap C_j=\varnothing(i\neq j)$。聚类过程可用公式（2-1）表示。

$$X_i\xrightarrow{\varphi}C_k(i=1,2,...N,k=1,2,...K) \tag{2-1}$$

在聚类算法中，数据样本间的相似性度量方法的优劣对聚类结果的好坏具有重要的影响，在实际生活中产生的数据绝大多数都是数值型数据，因此在聚类算法中针对数值型数据通常采用计算数据样本间的距离来对数据样本进行相似性度量。在一个 $N\times M$ 的数据集中，第 i 个数据样本和第 j 个数据样本分别表示为 $X_i=\{x_{i1},x_{i2},...,x_{iM}\}\in R^M$ 和 $X_j=\{x_{j1},x_{j2},...,x_{jM}\}\in R^M$，$X_i$ 和 X_j 之间的距离值可以表示为 $Dist(X_i,X_j)$，$Dist(X_i,X_j)$ 越小，X_i 和 X_j 间的相似度越大；反之，两者之间的相似度越小。聚类算法中通常采用的几种距离度量公式如下：

(1) 欧几里得公式（Euclidean Distance）

$$D(x_i,x_j)=\sqrt{(x_{i1}-x_{j1})^2+(x_{i2}-x_{j2})^2+\cdots+(x_{iM}-x_{jM})^2} \tag{2-2}$$

(2) 曼哈顿公式（Manhattan Distance）

$$Dist(X_i,X_j)=\left|x_{i1}-x_{j1}\right|+\left|x_{i2}-x_{j2}\right|+\cdots+\left|x_{iM}-x_{jM}\right| \tag{2-3}$$

(3) 闵可夫斯基公式(Minkowski Distance)

$$Dist(X_i,X_j)=\left(\left|x_{i1}-x_{j1}\right|^h+\left|x_{i2}-x_{j2}\right|^h+\cdots+\left|x_{iM}-x_{jM}\right|^h\right)^{\frac{1}{h}} \quad (h>0\wedge h\in \boldsymbol{Z}) \tag{2-4}$$

(4) 夹角余弦公式(Cosine Distance)

$$Dist(X_i,X_j)=1-\cos(X_i,X_j)=1-\frac{\sum_{m=1}^{M}(x_{im}\cdot x_{jm})}{\sqrt{\sum_{m=1}^{M}x_{im}^2\cdot\sum_{m=1}^{M}x_{jm}^2}} \tag{2-5}$$

欧几里得距离公式是在相似性度量研究中应用最为广泛的一种距离计算公式，这种距离计算方法是先计算数据样本之间对应属性值之差的平方和，然后再开方得到数据样本之间的距离，通过研究可以发现，欧几里得距离还具有如下三个特征[35]：

(1) 距离的非负性：$Dist(X_i,X_j)\geq 0$；若$Dist(X_i,X_j)=0$，则代表$X_i=X_j$；

(2) 距离的对称性：$Dist(X_i,X_j)=Dist(X_j,X_i)$；

(3) 距离的三角不等式：$\forall X_i$、X_j、$X_l\in R^M$，均有$Dist(X_i,X_j)\leq Dist(X_i,X_l)+Dist(X_l,X_j)$。

由欧几里得距离公式的三个特征可见，在使用欧几里得距离作为聚类算法中相似性度量标准时，要求同时满足以上三个条件。曼哈顿距离公式是利用数据样本之间对应的属性值之差的绝对值之和来度量数据样本间的相似度。闵可夫斯基距离公式是由欧几里得距离公式和曼哈顿距离公式演化而来的，是两种距离公式的推广。夹角余弦距离公式是通过计算两个数据样本点的向量夹角余弦值去度量数据样本点之间的相似性。

2.1.2 聚类方法分类

聚类在机器学习领域是一种无监督的学习方法，在应用中不是给计算机指定什么方法对聚类目标进行聚类操作，而是让计算机去学习选择怎样的方法，这样计算机往往能运行出比人为指定更好的聚类方法。目前已经研究出的聚类算法主要可以分为以下五类，它们分别是层次法、密度法、网格法、划分法、模型法[36 37]。

(1) 层次法：这一类算法是将需要进行聚类操作的数据集进行层次分解，当分解的结果满足设定的停止条件，分解结束，然后将最终的分解结果用树形图来表示。采用层次法的聚类算法主要采用凝聚的和分裂的两种层次聚类方式对数据集进行聚类。凝聚方式又称“自底向上”法，算法的初始化是将待聚类的每一个

数据样本分别看作一个类，通过逐层合并相似性较大的类，直到满足停止条件为止。分裂法又称“自顶向下”法，算法初始化是将待聚类数据样本归于同一个类，然后通过逐层分解，将大类分为不同的小类，直到满足停止条件为止。比较典型的基于层次的聚类算法有：BIRCH 算法、Chameleon 算法、CURE 算法等。

(2) 密度法：基于密度法是根据数据样本在空间分布的密度情况来进行聚类，这类算法与各种利用计算距离来进行相似度度量的算法相比较而言，它能最大程度克服因数据分布形状不同而对最终聚类结果带来的影响，基于距离算法一般适用于在空间分布成“球形”的数据，而基于密度法的算法可以搜索到任意形状的簇，通过判断某一中心点“邻域”的数据样本点是否超过算法设定的阈值来进行聚类，当某一数据样本分布区域中样本点分布的密度大于设定的阈值时，就将这一区域的数据样本点划分为一簇，若没有达到算法设定的阈值，则重新搜索，重新划分。比较典型的基于密度的聚类算法有： DBSCAN 算法、DENCLUE 算法、OPTICS 算法等。

(3) 网格法：网格法是将原始数据空间划分为有限个网格单元，然后以单个网格单元为对象对数据进行处理。与其他聚类算法相比较，采用网格法的聚类算法其优势不仅在于对数据的处理速率快，而且与数据样本点的数目无关，这一类算法只与数据空间中网格单元数目有关。比较典型的基于网格的聚类算法有：Sting 算法、Wave-Cluster 算法、Clique 算法等。

(4) 划分法：划分法首先是将给定的有 N 个数据样本的数据集初始化为 K 个划分（$K \le N$），一个划分为一类，每一个类必须满足两个条件：第一，每一个类里面至少包含一个数据样本点；第二，每个数据样本点只能属于一个类。然后通过相似性度量将数据样本点分到 K 个类中，通过不断的迭代更新，对初始的分类结果不断优化，最终达种全局的最优分类，即同类数据样本点间间隔越小越好，不同类间数据样本点间间隔越大越好。比较典型的基于划分的聚类算法有：PAM 算法、K-means 算法和 K-medoids 算法等。

(5) 模型法：这类算法是将数学中的模型应用于聚类中，算法对每一个类建立一个数学模型，然后寻找与每一个类模型最相拟合的数据样本点。模型是由一系列数学中的概率分布决定的，因此大多数是基于统计与神经网络两种方案，这两种方案因为都考虑到了数据样本中的“噪声”点和离散点，所以这类方法具有较好的鲁棒性。比较典型的模型聚类算法有：EM 算法、COBWEB 算法、SOM 算法以及 ART 算法等。

我们总结聚类算法的分类及其经典算法如图 2-1 所示：

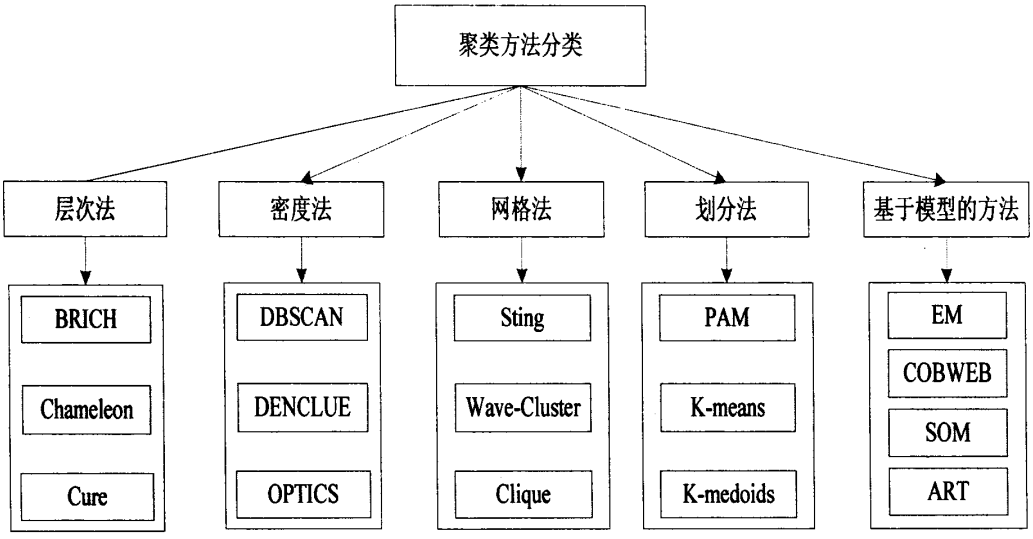


图 2-1 聚类算法分类与典型算法

一个好的聚类算法应该具有以下性质：

(1) 输入的参数少；

(2) 输入数据的顺序敏感度低；

(3) 有较好的可扩展性；

(4) 能处理噪声与多维数据。

聚类方法在高维性、适合的数据类型、时间复杂度等方面都有各自的优势和适用性，每一个聚类算法都不可能在每一个方面都优于其他的聚类算法，因此在实际应用中需要根据具体的应用环境对选择的聚类算法进行改进。聚类结果好坏，除了与所选算法有关之外，与算法的运行也有很大的关系，因为即使在同一个聚类算法中，在初始参数不同的情况下也会得到不同的聚类结果。即对于同一数据集，无论你使用同一种聚类方法还是使用不同的聚类算法，都可能得到不同的聚类结果。从上面可以看出，在实际实验中，无论怎么根据实际环境改进聚类算法，如果只是单纯依靠单个的聚类算法进行聚类，实验所得到的聚类结果往往都会存在较为严重的偏差，而且聚类结果也具有很大的不稳定性。为了改善聚类结果存在的偏差与不稳定性的情况，A. Strehl 和 J. Ghosh 提出了聚类集成这一概念。

2.2 聚类集成研究分析

Strehl 和 Ghosh 于 2002 年提出了聚类集成[38]的概念，聚类集成定义为：利用不同的聚类方法或同一方法不同初始值对原始数据集进行聚类，然后将多个基聚类结果进行合并处理。它的具体表述如下：假设有 N 个数据点的数据集 $X=\{x_1,x_2,x_3,.....x_N\}$，

对数据集进行 l 次聚类得到 l 个基聚类结果，$\prod=\{\pi_1,\pi_2,\pi_3\cdots\pi_l\}$ (以下称为基聚类成员)，其中 π_l 为第 l 次对原始数据集进行聚类得到的聚类结果。设计一种共识函数 Γ，对这 l 个基聚类成员进行合并，得到一个最终的聚类集成结果 C，通常情况下，最终的聚类结果是在 l 个基聚类结果中最好的。聚类集成的工作原理如图 2-1 所示。

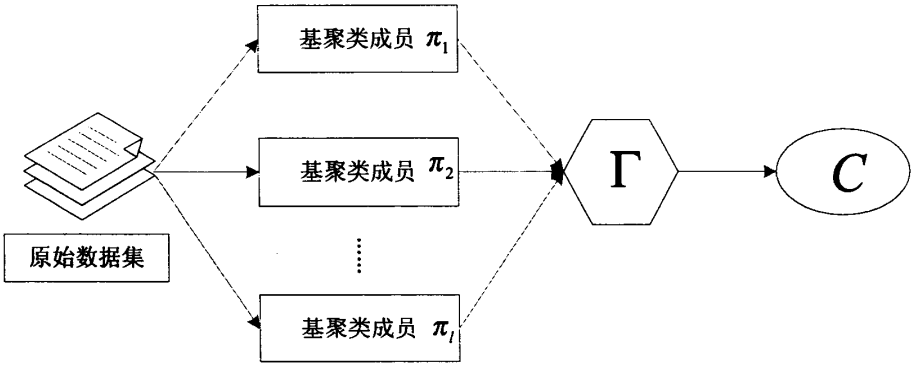


图 2-2 聚类集成工作原理图

因为在对基聚类结果进行集成的过程中结合了不同基聚类结果的特征，所以利用聚类集成方法对数据集进行聚类操作主要有以下几个优点[39]：

适用性：能产生比单个聚类算法更好的聚类结果。

鲁棒性：算法各个方面的平均性能更优越。

稳定性：稳定性较强，各种外部因素（噪声、孤立点等）对聚类结果的影响较小。

聚类集成算法中，要先产生对原始数据集聚类的 l 个基聚类成员，然后对这 l 个基聚类成员进行合并处理得到最终聚类结果。因此，聚类集成主要需要解决两个问题：

(1) 如何产生有效的基聚类成员，且不同的基聚类成员之间应该需要怎样的差异度。

(2) 如何设计共识函数，以便对基聚类成员进行合并，从而得到最终聚类结果。

图 2-3 展示了当前聚类集成的研究方法[40]。

由图 2-3 可知，产生基聚类成员的方法有多种，如：不同数据子集、同一算法不同的初始值、同一数据集不同的算法等。而在设计共识函数时,也有许多不同的方法，如：基于 Co-association 算法（包括 Single Link、Complete Link、Average Link 等 ）、投票算法、信息论算法、图论法(CSPA, HGPA,MCLA)和混合模型的 EM 算法等。

2.2.1 基聚类成员的产生

在产生基聚类成员方面，Strehl 等指出可通过不同的算法产生不同的基聚类成员。Fred 等[41]提出在数据样本中随机抽取几个数据点作为聚类中心点，然后通过多次运行

K-means 算法产生不同的基聚类成员。在现有的蚁群算法基础上，Yang 等[42]提出了

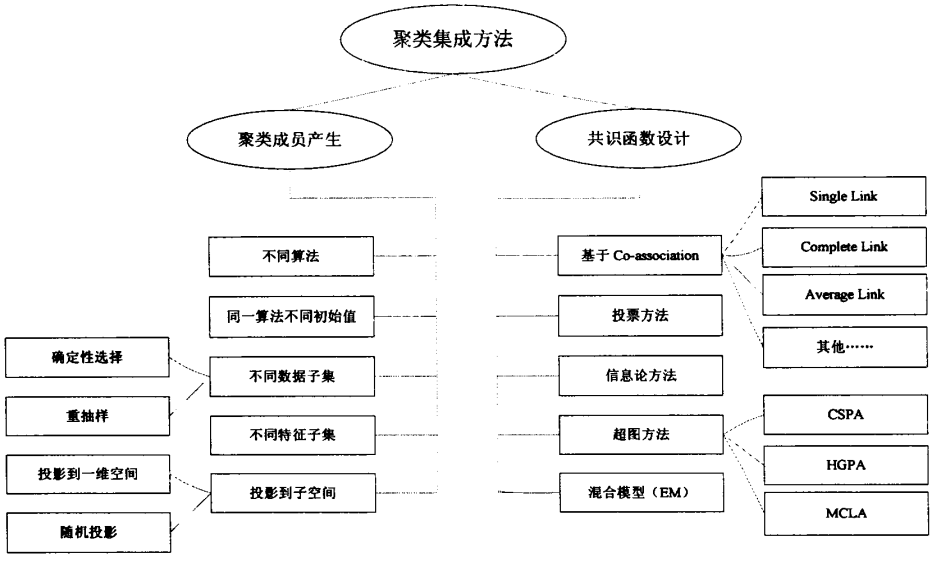


图 2-3 聚类集成研究方法图

一种改进的蚁群算法，这种算法是并利用蚂蚁不同的初始运动速度来产生具有差异性的基聚类成员。罗会兰[43]提出了一种基于数学形态学的聚类方法，该方法是将原本用于图像处理中的方法应用于聚类中，在找出数据中簇核心后再对这些簇核心进行集成操作，最终的聚类结果就是在这些簇核心集成的结果上得到。文献[44]提出了对原始数据集进行随机抽样，这些通过随机抽样得到的数据子集就可以从不同的层面将整个数据集的结构反映出来，这样可以让数据集的真实分布得到较好地揭示。然后利用 K-means 算法对随机抽样产生的数据子集进行多次聚类操作，从而产生多个不同的基聚类成员。文献[45 46]也是利用抽样的方式来产生数据子集，在这两篇文献里面采用的抽样方法是 Bootstrap 抽样法，利用抽样得到的不同数据子集来产生基聚类成员。文献[47]中使用综合的方法产生基聚类成员，如随机投影、随机投影与主成分分析、主成分分析与随机抽样等方法。受到监督提升算法的启发，A. Topchy 等人在文献[48]中设计了一个自适应的方案来生成基聚类成员。采用自适应方案产生基聚类成员会存在一个关于标志匹配的问题，同一个数据样本在先后的聚类操作中，分配到的聚类标签有可能不一样，为了解决标签匹配问题，文献中采用第一次聚类结果作为参考，然后利用 Hungarian 算法对标签进行匹配。通过实验结果对比证明这种自适应的方法能得到较好的聚类集成结果。通过对聚类集成多样性的研究，发现聚类集成中基聚类成员之间的差异度对聚类集成结果也有较大的影响[49]。在衡量基聚类成员间的差异度时，我们可以采用 Rand Index, Jaccard Index, Adjusted Rand Index, Mutual Information 等指标，这些指标的大小可以直接的反映出基聚类成员间差异度的大小。在后续的聚类研究过程中，S.T.Hadjitodorov 等人发现基聚类成员间的差异度大小与聚类集成质量

的好坏之间并非是成单调递增或者单调递减的关系。在一定的范围内，存在差异度可以使聚类得到很好的聚类集成结果，但是超过一定的范围，聚类集成结果的质量就会下降。利用基聚类成员间差异度大小来衡量聚类集成结果好坏的这种方法的缺点是不适合在大型矩阵数据中运用，在大型数据中应用，花费的时间比较大，文中提出可以引入阈值来改善此问题，经过分析发现，当基聚类成员间差异度阈值θ=0.5 时，得到的最终聚类结果优于采用其他阈值时的聚类结果。

从监督学习案例中得到启示，为了减少预测错误率，A. Topchy 等人[50]探讨了由“弱”聚类方法产生的基聚类成员对聚类集成结果的影响。“弱”聚类结果是指比采用随机划分法所得到的聚类结果好一些的聚类结果，该文献中提到有两种方法可以得到“弱”聚类结果，第一种方法是使用数据投影的方法，即将多维数据投影到一维空间；第二种方法是对数据进行切割，即用任意多个随机的超平面对数据进行切割，然后利用 K-means 算法对切割后得到的数据子空间进行聚类操作，从而得到基聚类成员。通过采用文献中所提到的集成方法进行实验，实验结果表明，利用“弱”聚类方法产生基聚类成员，最终得到的聚类集成结果比使用单一的聚类算法得到的聚类集成结果更好。

2.2.2 共识函数的设计

用来将若干个基聚类成员进行合并的函数称为共识函数，在 A.Strehl 和 J.Ghosh 提出聚类集成这一概念之前，已经有研究人员对不同的聚类结果进行合并的研究，期望合并后的聚类结果比单一的聚类结果更优。在文献[38]中介绍了三种有效的基于超图法的集成方法，分别是 CSPA、HGPA、MCLA，这三种集成方法都是通过将划分集合转换成为超图的表示形式来对基聚类成员进行操作的。下面简单介绍一下这三种集成方法：（1）CSPA（(Cluster-based Similarity Partitioning Algorithm），这种集成方法是给每一个基聚类成员都生成一个$h \times h$（行数、列数都为h）的二元相似矩阵，然后判断两个数据对象是否属于同一个类中，如果两个数据对象属于同一个类，那么矩阵中该元素的值为 1，如果两个数据对象不属于同一个类，则该元素在矩阵中的值为 0。（2）HGPA（Hypergraph Partitioning Algorithm），这种集成方法是将每一个数据当成顶点，把聚类成员中的每个类当成是一条超边，然后分别将每一条超边，所有的顶点都分别看作是具有相同权重的超边和顶点，最后利用超图划分法 HMETIS 算法对这些基聚类成员进行集成操作；（3）MCLA (Meta-Clustering Algorithm)，这种集成方法是以基聚类成员为基础，把聚类成员中的每个类当成是一条超边，然后将超边进行分组，再把每个数据对象分配到分裂的超边中，最后根据数据点在超边中出现的比例来判定其最终归于哪一类。

Fred 提出了 Voting-K-means 算法，该算法是利用随机产生初始点进行 K-means

算法运算，运行ℓ次得到基聚类成员，然后再计算出基聚类成员间的 Co-association 矩阵。Co-association 矩阵方法[51]是用于衡量矩阵中数据点之间的相似度。例如基聚类成员中的第i个数据点与第j个数据点之间的相似度可以用如下式子表示：

$$A_{ij}=\frac{i\text{和j属于同一类的次数}}{\text{聚类算法运行总次数}\ell} \tag{2-6}$$

该方法中采用 0.5 作为衡量数据点间相似度的阈值，即在 Co-association 矩阵中，如果有一半以上的基聚类成员认为某两个数据点是属于同一类，则算法就把这两个数据点归为同一类中。阈值的选取也需要根据实际经验，有时候阈值选取错误，得到的结果就会产生很大的误差，所以阈值的选取还需要进一步研究。Topchy 等在文献[52]中为共识函数建立了一个关于基聚类成员间的概率模型，通过使用 EM 集成算法对基聚类成员进行集成操作。Zhou 等[53]根据 K-means 算法中初始中心点为随机选择的原理，将 K-means 算法重复运行多次，以此来产生不同的基聚类成员，然后利用基聚类成员间的互信息值来给各个数据点设定不同的权重，最后再通过采用投票法，加权投票法，选择投票法以及加权/选择投票法四种方法对基聚类成员进行集成操作。Yang 等[54]受到神经网络集成算法思想的启发，设计了一种基于自适应谐振理论的集成算法，该算法能够通过模拟人的大脑处理问题的方式来完成规律总结等过程。Luo 等[55]将聚类集成技术应用于无监督的特征选择中，该方法从基聚类成员间的共联矩阵中获取到各个数据样本特征之间的差异度信息，然后采用一种改进后的 Relief 算法对这些特征进行评估，最后通过评估的结果来对数据样本进行聚类操作。李杉等[56]提出一种基于 Bagging 的聚类集成方法，该方法通过利用一种新的数据采样技术来产生不同的数据子集，利用这种方法产生的数据子集，不仅能够产生具有多样性的数据子集，而且能够保证这些数据子集间具有最大相关性，然后采用一种改进后的 K-means 聚类算法对这些数据子集进行聚类操作，产生多个基聚类成员，根据这些基聚类成员间的互信息的大小对数据集归类处理，最后对初次划分具有争议性的数据对象通过计算其与各个聚类中心点的距离，根据到不同中心点距离的大小将这些具有归类争议的数据对象重新划分到新的类中。Jugurta Montalvão 等[57]在分析了共识函数准则的基础上，提出了一种新的利用基聚类成员间的多样性来进行集成的方法。EA (Evidence Accumulation)算法[58]是通过计算数据对象间的 Co-association 矩阵，再利用基于最小生成树 MST(Minimum Spanning Tree)的分级算法在 Co-association 矩阵的基础上进行聚类集成操作。此后 A. Fred 等又改进了以上方法：在一定的取值范围内确定一个类别数目 K 值，然后再在已有的基聚类成员中利用 K-means 算法将这些数据对象最终聚为 K 个类[59]。

JSDCC（Jensen- Shannon Divergence based Clustering Combination)算法[60]是利用

信息论度量的方法来建立一个共识函数模型，该方法从数据对象间的Co-association矩阵中提取出一组联合分布模型，通过计算矩阵中数据点与模型的Jensen-Shannon值，然后根据Jensen-Shannon值的大小来最终决定数据对象的最终所属类别。这个算法中的关键点是模型化近邻数据对象的关联性，因此利用这个算法来对基聚类成员进行集成时，解决了聚类问题中对于初始值敏感的问题。文献[61]的研究人员在CSPA算法的基础上做了进一步的扩展，提出了WSnnG (Weighed Shared nearest neighbors Graph)算法。此算法是将数据对象间的Co-association矩阵稀疏化来得到聚类集成的基聚类成员，在该文中通过保留每个数据点与其邻近点的联系，然后以稀疏后的矩阵生成图，再用基于图论的集成算法来得到聚类集成结果。此后又有研究人员改进了WSnnG方法[62]，该方法是给算法指定一个K值，然后在生成图的时候只利用每一个数据点邻近的K个数据点，这样就大大简化了生成图的规模，而且使得算法的计算量也得到极大的减少，最终的实验结果表明这样得到的聚类集成结果与之前的聚类结果无明显的变化。

在文献[63]中，研究人员通过把每一个数据对象的聚类结果看成是数据集的一个特征，这样就把数据对象对应地映射成数据点的 M 维属性（数据集的第 i 个数据对象所得到的聚类结果就用第 i 维的值来表示），然后用 Kerouac 方法在数据集映射后的属性空间进行聚类,得到最终的聚类结果。该算法不用预设聚类的类别数，且其扩展性较好。HBGF (Hybrid Bipartite Graph Formulation)算法[64]主要是生成一个双元图模型，该图模型中既包含了数据对象，也包含了类，用这个双元图模型进行聚类集成操作时，利用了 METIS 算法对生成的双元图进行基于图论的聚类，在产生最终聚类结果过程中考虑了不同数据对象之间和不同类之间的相似性。

以上内容总结了现阶段聚类集成技术在基聚类成员的产生和设计共识函数研究方面所取得的成果。现阶段聚类集成技术已经得到了广泛的发展，但是还没有达到十分成熟的状态，所以在今后的聚类集成技术研究中，关于基聚类成员的产生和共识函数的设计方面还需要有以下三方面的突破：

(1) 在基聚类成员产生方面，如何产生基聚类成员，从而使最终的聚类结果达到最优化。

(2) 在共识函数设计方面，需要增强集成算法的可扩展性以及集成算法在应用过程中的增量式研究。

(3) 在实际应用方面，聚类集成已经在医学诊断、人工智能等方面应用，因此如何将聚类集成技术应用到更广泛的实际生活中将有重要的意义。

2.3 本章小结

本章主要介绍了有关聚类方面的一些基本概念、研究方向和现阶段已有的研究成

果。本章开始阶段详细介绍聚类的概念及其定义，并详细介绍了聚类算法在利用数据样本点间的距离大小对数据样本点进行相似性度量时采用的几种距离公式，然后介绍了现阶段已有的聚类算法分类情况。本章的最后对聚类集成进行了详细的阐述，具体对在聚类集成过程中已有的基聚类成员的产生方法以及集成过程中共识函数的设计方法进行了详细介绍。

第 3 章 基于最小冗余特征子集的子空间聚类集成研究

3.1 变量之间的相关关系

在进行最小冗余法划分子空间的研究的过程中，我们把数据的每一个属性看成是一个变量，判断数据属性间的关系，即判断不同变量之间的关系。

变量之间的关系通常分为两种：一种为确定性的函数关系，即一个变量由另一个变量或一组变量完全确定；另一种为不同变量之间存在某种关系，但是这种关系不具备确定性，而是带有随机性的，这种情况我们称两变量之间具有相关关系。这两种变量之间的关系的异同点我们可以归纳如下：

(1) 两种关系的相同点：这两种关系都是指变量之间的关系。

(2) 两种关系的不同点：首先，函数关系是一种确定性的关系，而相关关系是一种非确定性的关系；其次，函数关系是一种因果关系，而相关关系却不一定是因果关系，有可能是伴随关系。

3.1.1 变量之间的相关关系的衡量标准

为了确切地衡量出两个随机变量之间的关联程度，统计学家皮尔逊提出了相关系数这一概念。相关系数是用来反应两个随机变量之间关系密切大小的一个指标。相关系数的计算是按照积差的方法来进行的，同样也可以根据两个随机变量与它们各自的平均值的离差，最后通过两变量的离差相乘来反映两个变量之间的关联程度。衡量随机变量相关性的方法主要有三种：

(1) 皮尔逊(Pearson)相关系数：皮尔逊相关系数被广泛用于衡量两变量之间的相关程度，它的取值范围介于-1 到 1 之间。皮尔逊相关系数与其他相关系数不同的时它有一个重要的数学特性：如果两个变量的位置和尺度发生改变，这些改变都不会引起皮尔逊系数的变化，例如我们把回归直线方程中 $y=a+bx$ 中 x 和 y 进行移动，使方程变为 $x=c+dy$，a,b,c,d 均为常数，其中变量 x 和 y 的位置改变了，但这样的改变并不会引起常数的变化。皮尔逊相关系数的计算公式如下。

$$r=\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2\cdot\sum_{i=1}^{n}(y_i-\bar{y})^2}}=\frac{n\sum_{i=1}^{n}x_iy_i-\sum_{i=1}^{n}x_i\cdot\sum_{i=1}^{n}y_i}{\sqrt{n\sum_{i=1}^{n}x_i^2-(\sum_{i=1}^{n}x_i)^2}\cdot\sqrt{n\sum_{i=1}^{n}y_i^2-(\sum_{i=1}^{n}y_i)^2}} \quad (3\text{-}1)$$

(2) 斯皮尔斯(Spearman)相关系数：斯皮尔斯相关系数是衡量两变量在相互依赖性方面的非参数指标。斯皮尔斯等级相关系数是利用单调方程来评价两个统

计变量的相关性。斯皮尔斯相关系数可以由计算皮尔逊相关系数的方法来计算，计算过程中只需要把原随机变量中的原始数据替换成其在随机变量中的等级顺序即可：例如：（1,5,13,20）可以替换成（1,2,3,4），而（39,20,15,26）可以替换成（4,2,1,3），然后再求被替换后的两个随机变量的皮尔逊相关系数即可。当数据中没有重复值且两变量完全单调相关时，斯皮尔斯相关系数就为-1 或者+1。

(3) 肯德尔（Kendall）相关系数：肯德尔相关系数又被称为和谐系数，是一种等级相关系数。该系数的计算方法为：变量 x, y 的两对观察值 $X_i Y_i$ 和 $X_j Y_j$，如果 $X_i < Y_i$ 且 $X_j < Y_j$，或者 $X_i > Y_i$ 且 $X_j > Y_j$，我们就称这两对观察值是和谐的，否则这两对观察值就是不和谐的。肯德尔相关系数的计算公式如下。

$$\tau = \frac{(number\ of\ concordant\ pairs) - (number\ of\ discordant\ pairs)}{\frac{1}{2}n(n-1)} \qquad (3\text{-}2)$$

即在变量的所有观察值对中（总数为：$0.5*n*(n-1)$ 对），和谐的观察值对减去不和谐的观察值对的数量，除以总的观察值对数。

相关系数的类型可以根据研究对象的不同划分为：简单相关系数，复相关系数和典型相关系数。简单相关系数又被称为线性相关系数，是被研究得最多的一种相关系数，它是用来度量两个变量间的线性关系；复相关系数又被称为多重相关系数，其中复相关的含义指的是因变量与多个自变量之间的相关关系；典型相关系数是对原来各组变量先进行主成分分析操作，从而得到变量间新的线性关系的综合指标，最后以得到的综合指标之间的线性相关系数为依据，再倒推回去研究原来各组变量间的相关关系。

相关系数的取值范围是在-1 到 1 之间，如果是变量 x 和 y 之间的相关系数，当相关系数的绝对值为 1 时，称变量 x 和 y 完全相关，且此时变量 x 和 y 之间具有线性函数的关系，当相关系数为 0 时，称变量 x 和 y 不相关。当相关系数绝对值小于 1 时，变量 x 变动引起变量 y 值的变动，相关系数的绝对值越大，由变量 x 变动引起的变量 y 的变动越大，其中，当相关系数绝对值大于 0.8 时，称两变量为高度相关，当相关系数绝对值大于 0.5 小于 0.8 时，称两变量为显著相关，当相关系数绝对值大于 0.3 小于 0.5 时，称两变量为低度相关，当相关系数绝对值小于 0.3 时，则称两变量无相关。

在前面介绍的几种相关系数中，本次论文研究的主要是简单相关系数，对应的即是研究两变量之间具有的线性相关性研究。

3.1.2 变量间线性关系的衡量

变量间的线性相关性指的是具有相关关系的两变量，如果它们的所有观察数据点都分布在一条直线附近，那么我们就称这样的两变量之间具有线性相关关系。变量间

的线性关系可以分为正相关与负相关两种关系。如果两个变量之间的关系是其中一个变量值由小变大时，另一个变量也随之由小变大，我们就把变量之间具有的这种线性关系称之为正相关；相反，如果两个变量之间的关系是一个变量由小变大，另一个变量随之由大变小，我们就把变量之间具有的这种线性关系称之为负相关。在具有线性相关关系的变量之间，可以划许多条在观察值数据点附近的线，其中最靠近这些数据点的一条称之为回归直线。回归直线描述的是变量的观察值数据点之间接近的程度，直观反映出了变量之间相互影响的关系，因此，我们可以通过回归直线方程来描述变量之间的关系或者做一些变量的回归分析。

回归直线可以通过利用变量之间的散点图来得到，设回归直线方程为 $y=a+bx$，其中 a,b 为回归系数，则求解回归直线方程的步骤为：

(1) 作出给定数据集的散点图，直观地判断变量之间是否为线性相关；

(2) 求出 $\bar{x},\bar{y}$；

(3) 求出 x_i^2， x_iy_i；

(4) 求出 a,b，写出回归直线方程。

其中回归直线方程系数 a,b 可以用最小二乘法求得：

$$b=\frac{\sum_{i=1}^{n}x_iy_i-n\bar{x}\bar{y}}{\sum_{i=1}^{n}x_i^2-n\bar{x}^2}=\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2}\quad,\quad a=\bar{y}-\bar{b} \tag{3-3}$$

其中 $\bar{x}=\frac{1}{n}\sum_{i=1}^{n}x_i$， $\bar{y}=\frac{1}{n}\sum_{i=1}^{n}y_i$。因为回归直线是最贴近所有观察值数据点的直线，因此我们可以用离差的平方和来反映变量之间的贴近程度，即总离差 $Q=\sum_{i=1}^{n}(y_i-a-bx_i)^2$，回归直线的总离差值是散点图里所有直线中最小的那一条。

在研究变量之间的线性关系过程中，利用求得的回归直线方程可以帮助我们探寻事物发展的某些规律，预测一些较难得到的数据，以此来为我们做判断提供依据。但是在利用回归直线方程的过程中，我们需要注意到我们做的回归预测分析前要做散点图，以此判断是否是线性相关关系，且做的回归预测分析要有实际意义。

3.2 数据冗余的分类

数据冗余的定义：数据在一个数据集合中重复出现的现象就被称为数据冗余。其表现形式可以分为两种，第一种是数据之间的重复，第二种是将同一数据存放在不同的数据文件中。现阶段数据冗余的种类一般可以分为以下六个类：空间冗余、时间冗余、信息熵冗余、结构冗余、知识冗余以及视觉冗余。在实际研究过程中，得到较为广泛研究的数据冗余种类主要有空间冗余，时间冗余，结构冗余以及视觉冗余这四种类型。这四种冗余类型具体介绍如下：

(1) 空间冗余

这一类型的数据冗余在图像数据中经常出现。在图像中，表面颜色是有序分布的物体，其表面物理体征都是具有关联性的，那么在数字化图像过程中，这些具有关联性的光成像结构就表现为数据冗余。

(2) 时间冗余

这一类型冗余经常包含于序列图像和语言数据中。在序列图像中，前后紧邻的两张图像之间存在着较大的相关性，这样就表现为时间冗余。人们在说话的时候，发音的过程在时间上并不是完全独立的，而音频又是一个连续渐变的过程，因此在语言中同样也存在时间冗余。

(3) 结构冗余

一些图像在整幅图中的较大区域上都有着比较清晰的纹理结构分布，我们把这种分布模式称为结构，那么结构冗余就是指在图像中重复出现或者相近的纹理结构。

(4) 视觉冗余

人类的视觉感官并不能对图像场任何的细微变化都做出反应。例如我们对图像进行压缩操作后，再解压出来的图像其实已经在原始图像上发生了细微的变化，正是因为这些细微的变化在我们视觉系统所能辨别的范围之外，所以我们仍然认为图像很好的。人类的视觉系统对色度的分辨力约为 2^6 灰度级，但是大多数图像量化都是采用的 2^8 灰度级，我们就把这类冗余称之为视觉冗余。

在现实的社会活动中，通常为了达到某一种目的而采用数据冗余，例如：企业为了简化流程，给企业下属的各个部门发送同样的信息，将数据进行冗余性编码来防止数据的丢失、错误。数据冗余是人们在生活学习中所必然产生的东西，它虽然没有绝对的好与坏，但是数据的冗余会妨碍数据的完整性以及造成计算机存储空间的浪费，因此如何在增加数据独立性的同时减少数据冗余是现阶段数据应用的研究热点。

3.3 基于最小冗余子空间划分的研究

如何从原始的高维数据中有效地划分出最具代表性的数据子空间来进行子空间聚类，是子空间聚类集成研究中一个热点问题。合理有效地划分数据子空间，能够提升数据聚类集成的准确率。假设给定数据集 X 有 N 个样本点和 M 维属性特征，即 $X=\{x_j, j=1,...,M\}$，划分子空间就是要从数据的 M 维形成的空间 R^M 中选择出维数低于 M 的子空间 R^m。原始数据集 X 一共组成 2^M 个子空间，空间维数不大于 M 的子空间数目是 $\sum_i^m \binom{M}{1}$，因此在高维数据中，如何合理有效地搜索划分特征子空间是一项很艰巨的工作任务。一些基于近似方案的顺序搜索法陆续被提出来，包括最优单特征法[65]、顺序向前搜索法[66]、顺序向前浮动搜索法[67]等。

在无监督的情况下，采用有效的子空间划分法来对原始数据进行子空间划分，是

提升数据聚类准确率的关键。本次毕业设计提出最小冗余法来对数据集进行子空间划分。因为数据特征变量之间存在着广泛的联系，最小冗余法是通过计算数据特征变量之间的互信息大小，然后将相关性大的特征变量划分在一起。特征变量之间互信息值越大，说明两变量之间关联程度越大，反之，特征变量之间互信息值越小，说明两变量之间关联程度越小。假设给定两个随机变量x,y，两变量之间的互信息值大小由概率密度函数$p(x),p(y)$和$p(x,y)$共同决定，变量间互信息值计算公式如下：

$$I(x,y)=\iint p(x,y)\log\frac{p(x,y)}{p(x)p(y)}dxdy \tag{3-4}$$

通过计算互信息值来划分数据的子空间，可以确保子空间内变量之间具有最大的相关性，而变量之间的冗余最小。

3.3.1 变量间依赖关系研究

计算变量间的互信息，就是为了搜索到有m维的特征变量$\{x_j\}$组成的子空间S，搜索到的子空间S与我们的目标聚类结果c间具有最大依赖性，我们可以将关系表示如下：

$$\max D(S,c),\quad D=I(\{x_j,i=1,...,m\};c) \tag{3-5}$$

当$m=1$时，得到最大$I(x_j;c)(1\le j\le M)$.当$m>1$时，一种简单的增量搜索方法是每搜索一次增加一维数据：$m-1$维特征变量组成子空间S_{m-1}，当增加到第m维特征变量时，我们可以得到互信息的最大增量值$I(S;c)$，公式 3-6 如下：

$$\begin{aligned}I(S_m;c)&=\iint p(S_m,c)\log\frac{p(S_m,c)}{p(S_m)p(c)}dS_m d_c\\&=\iint p(S_{m-1},x_m,c)\log\frac{p(S_{m-1},x_m,c)}{p(S_{m-1},x_m)p(c)}dS_{m-1}dx_m d_c\\&=\int\cdots\int p(x_1,...,x_m,c)\log\frac{p(x_1,...,x_m,c)}{p(x_1,...,x_m)p(c)}dx_1...dx_m d_c.\end{aligned} \tag{3-6}$$

我们可以利用最大依赖性得到理论上最大互信息值，但是在实际研究中，因为在高维空间中数据样本是有限的，再加上多元密度估计往往涉及到高维协方差矩阵的逆运算，而这个运算通常是一个不太适定的问题，因此要得到多元密度函数$p(S_{m-1},x_m)$和$p(S_{m-1},x_m,c)$是比较困难的。

3.3.2 变量间最大相关性和最小冗余研究

依据变量间最大依赖性原理在实际研究中很难实现，因此可以考虑基于变量间最大相关性来衡量特征变量之间的关联关系。变量间最大相关性原理是通过公式（3-7）来进行特征变量选择，从而实现对子空间的划分，变量间最大相关性原理的公式近似于（3-5）中的$D(S,c)$，D是由所有单个特征变量与类c之间互信息总和的平均值：

$$MaxD(S,c),\quad D=\frac{1}{|S|}\sum_{x_i\in S}I(x_i;c) \tag{3-7}$$

虽然利用特征变量间的最大相关性原理可以实现对数据特征变量的选择，但是我们可以发现利用这个原理来选择的特征变量之间会存在大量的冗余。为了实现两个相关度很高的变量在他们其中某一个变量被删除或者漏选之后，不会引起每一个变量的辨识度有太大的改变，我们在特别变量选择的时候引入最小冗余，表达式如下所示[68]：

$$\min R(S),\quad R=\frac{1}{|S|^2}\sum_{x_i,x_j\in S}I(x_i,x_j) \tag{3-8}$$

我们定义一个变量$\Phi(D,R)$结合变量D,R，也用如下表达式来同时优化变量D,R：

$$\max\Phi(D,R),\quad \Phi=D-R \tag{3-9}$$

在实际研究中，增量搜索方法可以用来寻找到由$\Phi(.)$所定义的邻近最优特征变量。假设我们已经搜索到由$m-1$维特征变量组成的空间S_{m-1}。如果要从剩余的特征变量$\{X-S_{m-1}\}$中搜索到第m维特征变量，我们就可以通过寻找能够使得$\Phi(.)$最大化的特征变量来确定第m维特征变量，其关系式可以由如下式子表示：

$$\max_{x_j\in X-S_{m-1}}\left[I(x_j;c)-\frac{1}{m-1}\sum_{x_i\in S_{m-1}}I(x_j;x_i)\right] \tag{3-10}$$

由以上描述我们可以得知，利用最小冗余法来进行特征变量选择的计算复杂度为$O(|S|\cdot M)$。

3.3.3 基于变量间最小冗余的子空间划分法

基于变量间最小冗余法的第一步是计算数据特征变量之间的互信息。在划分子空间的时候，可以采用多种方法来进行划分，比较常见的方法是穷举法，即根据第一步计算得到的数据中每一对特征变量之间的互信息值的大小将每一个特征变量进行分类，以此来实现子空间的划分。但是这种方法的开销比较大，因此在本论文中，我们将采用改进的 K-means 算法来实现基于特征变量间最小冗余的子空间划分。

传统的 K-means 算法是一种基于距离的无监督聚类算法，其中心思想是将每一个数据样本点分配到与某一类的中心点距离最近的类。K-means 算法的目标函数如下：

$$J_{K-means}=\sum_{j=1}^{k}\sum_{x_i\in C_j}\left\|x_i-u_j\right\|^2 \tag{3-11}$$

公式中u_j代表样本点x_i所属的类C_j的中心点。$J_{K-means}$是数据样本点到其对应类中心点距离的平方和，K-means 算法的目标就是要使得$J_{K-means}$的值越小越好。K-means 算法的流程如图 3-1 所示。

输入：聚类个数 K，数据集合 $X=\{x_0,x_1,...,x_{N-1}\}$；
输出：聚类划分 $C=\{c_0,c_1,...,c_{K-1}\}$；
1.随机选择 K 个数据样本作为初始的聚类中心点，如 $c_0=x_0,c_1=x_1,...,c_{K-1}=x_{K-1}$；
2.分别计算 $x_0,x_1,...,x_{N-1}$ 与中心点 $c_0,c_1,...,c_{K-1}$ 的距离，若与 c_i 的差值最小，则将其类标标记为 i；
3.对于所有类标为 i 的数据实例，通过计算其平均值作为新的中心点 c_i；
4.重复 2、3 步骤，直到 c_i 值的变化小于给定阈值或迭代次数达到最大值。

图 3-1 k-means 聚类算法流程

本论文利用改进的K-means算法来进行子空间划分，将计算数据特征变量间的互信息值替换k-means算法中计算数据样本点间的距离，算法的最终输出结果即为我们利用最小冗余原理所划分的子空间，划分出来的子空间就是数据中具有最小冗余的特征子集（minimum redundancy feature subset：MRFS），MRFS子空间划分法的具体过程如下：首先从数据特征变量中任意选择 K（K 为划分子空间个数）个特征变量作为初始目标特征变量，然后将剩余的特征变量与选出来的目标特征变量分别计算互信息，根据互信息值的大小，分别将特征变量分配到与其最相似（互信息值最大）的类中，完成一次特征变量的迭代，就计算一次每一个类别中特征变量间互信息值的总和，然后取每一类中与原目标特征变量互信息值最大的特征变量为新的目标特征变量，然后进行第二次特征变量迭代，第二次迭代结束后计算每一个类别中特征变量间互信息值的总和，将第二次迭代得到的互信息值总和的大小与上一次得到的互信息值总和相比较，不断重复这一过程，直到后一次互信息值总和与前一次互信息值总和的差在一定范围内就停止迭代。最小冗余法的目标函数定义如下：

$$H=\sum_{j=1}^{k}\sum_{x_i\in G_j} I\left(x_i;v_j\right) \tag{3-12}$$

公式中 v_j 代表特征变量 x_i 所属类 G_j 的目标特征变量。H 代表每一类中特征变量之间互信息值的总和。最小冗余法就是要使 H 的值越大越好。其算法流程如图 3-2 所示。

在利用最小冗余法对数据进行子空间划分的过程中，阈值的选取十分关键，选取的阈值合适与否，直接决定着子空间划分结果的好坏。本次论文中阈值不是统一的选取某一个数值来充当阈值，而是根据实验数据集规模的大小来选取。当实验数据规模大时，各子空间中特征变量间互信息中的总和比较大，我们在设定阈值时就需要选取大一些的值，当实验数据规模小时，各子空间中特征变量间互信息中的总和比较小，我们在设定阈值的时就需要选取小一些的值，这样根据数据规模来动态的设定实验阈值，能够在一定程度上改善最终的实验结果。

输入：子空间划分个数 K，数据特征变量集合 $X=\{x_0,x_1,...,x_{M-1}\}$；

输出：子空间划分 $G=\{G_0,G_1,...,G_{K-1}\}$；

1.随机选择 K 个特征变量作为初始目标特征变量，如 $G_0=x_0,...,G_{K-1}=x_{K-1}$；

2.分别计算 $x_0,x_1,...,x_{M-1}$ 与初始目标特征变量 $G_0,...,G_{K-1}$ 的互信息值，若与 G_j 的互信息值最大，则将其类标标记为 j；

3.对于所有类标为 j 的数据特征变量，计算它们与目标特征变量的互信息值总和，并计算后一次与前一次互信息值总和之差，然后取每一类中与原目标特征变量互信息值最大的特征变量为新的目标特征变量；

4.重复 2、3 步骤，直到前后两次子空间划分中特征变量间互信息值总和之差小于给定阈值就停止迭代运算。

图 3-2 MRFS 算法流程

结合 MRFS 算法的流程，我们可以得到基于最小冗余子空间划分法的流程图如图 3-3 所示。

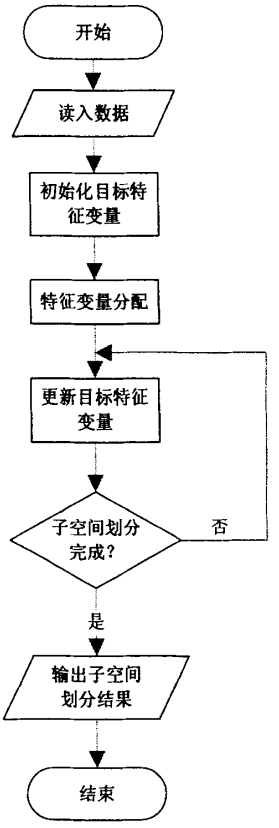


图 3-3 MRFS 算法划分子空间流程图

当 MRFS 算法运行结束后，输出的是对数据集进行子空间划分的结果，我们把这些划分结果作为后续实验的数据输入。

3.3.4 基于最小冗余的子空间聚类集成原理图

通过最小冗余法对数据的特征变量进行子空间划分，利用子空间划分的结果，利用聚类算法对划分结果进行聚类得到基聚类器，再通过选取共识函数对基聚类器进行集成，得到最终的聚类集成结果。因此基于最小冗余特征子集的子空间聚类集成的工作原理图如图 3-4 所示：

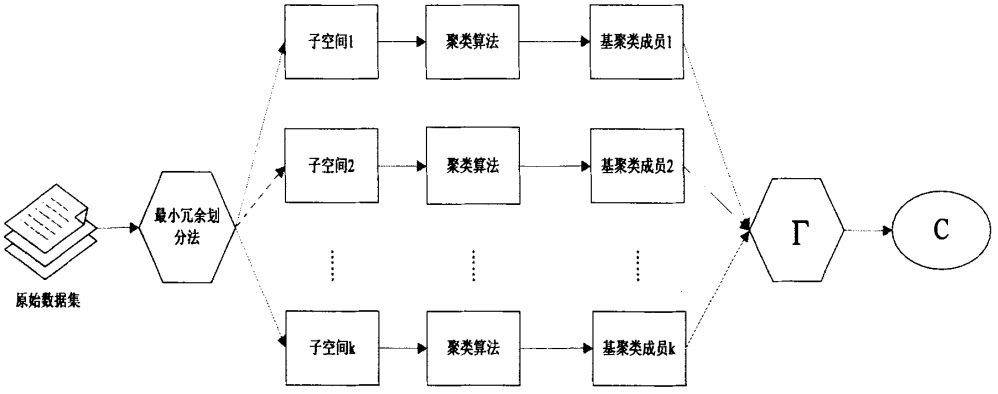


图 3-4 MRFS 算法聚类集成工作原理图

3.4 本章小结

本章主要介绍了利用最小冗余法来对数据进行子空间的划分。本章首先介绍了变量之间相关关系，衡量变量间相关性的三种常用相关系数的特点以及如何利用回归直线方程来反应变量间的数量关系。其次本章介绍了数据冗余的分类情况以及由数据冗余所带来的影响。随后本章重点介绍了如何基于最小冗余对数据特征变量进行子空间的划分，我们可以通过计算特征变量之间的互信息值，并采用增量搜索的方法来进行特征选择从而达到子空间划分的目的，但是这种方法在实际实验中有一些参数是很难得到的，所以我们通过利用变量间最大相关性，并加入最小冗余来限制特征变量的选择，最终实现对数据特征变量子空间的划分。本章的最后总结了基于最小冗余子空间划分的流程，并得到基于最小冗余特征变量的子空间聚类集成的工作原理。

第 4 章 基于属性最大间隔的子空间聚类

4.1 MMSC 算法介绍

已有的子空间聚类算法绝大多数都是根据数据点的密度以及数据点间的距离来划分子空间，通过对已有子空间聚类算法的分析，我们发现已有的子空间聚类算法具有一定的缺陷：(1)运行算法时需要输入的参数比较多；(2)算法对数据结构有一定的要求；(3)对噪音数据点的处理效果不理想。为了改善已有算法的缺陷，本文提出了基于属性最大间隔的子空间聚类（Maximum Margin Subspace Clustering ：MMSC)算法。MMSC 算法采用的是数据属性间的最大信息系数来对数据属性间的关系进行衡量，因此 MMSC 算法具有如下三点优势：

(1) MMSC 算法可以处理混合属性。

(2) MMSC 算法首次提出利用数据属性间的最大信息系数来对数据属性间间隔的判断，数据属性间的最大信息系数值越大，属性间间隔越小，反之，属性间间隔越大，最后利用最大间隔原理来进行子空间划分。

(3) MMSC 算法通过计算各属性间的互信息值来判断各属性间关联度的大小，因为数据在高维空间中分布比较稀疏，一些重要的数据属性在以往的子空间划分法中很容易被错误的划分，利用 MMSC 算法计算各属性间的互信息值可以避免以往算法对一些分布较散的重要属性的错误划分，能最大限度的减少有效信息的丢失，从而保持数据的原有特性。

4.1.1 属性间最大信息系数

衡量变量之间关联度的大小可以通过计算变量之间的互信息来判断，但是互信息法的缺点是它只能处理离散数据之间的关系，因此想要衡量混合类型数据之间的相关性就需要寻找其他的方法。2011 年，David N. Reshef 等人提出 MIC 算法(maximal information coefficient)[69],MIC 算法是一种新的寻找大数据集中数据属性之间关联关系的方法。MIC 算法有如下两点优势：

(1) 能对混合类型的数据进行处理。MIC 算法除了能够对本身是离散型的数据进行处理以外，还能够通过对连续型数据进行离散处理，从而实现对混合类型数据的处理。

(2) 能够更精确的表示数据间的关系。MIC 算法通过构建互信息特征矩阵来寻找变量之间的最大信息系数，因此可以很精确的表示出数据属性间关联性的大小。

MIC 算法原理是通过计算一个数据集中两两变量之间的互信息，通过将互信息值

归一化后构建一个特征矩阵$G(\chi)$，然后搜索出每两个变量之间的最大相关系数。

互信息是衡量两个变量之间的相关性大小的物理量，两个变量之间的互信息越大，说明两个变量之间的相关性越大，反之，则两个变量之间的相关性越小。设原始数据集N个数据点，有M个数据属性，即$\chi = \{x_{.,1}, x_{.,2}, \ldots, x_{.,M}\}$。将这$M$个数据属性划分为$K$类，则数据集$\chi$的属性可由$K$个簇$\{c_l \mid l=1,\ldots,K\}$表示。$f_i$和$f_j$是数据集中两个属性，我们用$I(f_i,f_j)$表示$f_i$和$f_j$之间的互信息，$H(f_i)$表示变量$f_i$的熵。$I(f_i,f_j)$就是我们的度量标准。$I(f_i,f_j)$的值是没有上界限的，因此我们可以比较容易地根据$I(f_i,f_j)$值的大小去解释和比较变量之间相关性。标准化的$I(f_i,f_j)$值的范围是在 0 到 1 之间。已有的几种标准化方法是以$I(f_i,f_j) \le \min(H(f_i),H(f_j))$为依据的。这些方法包括使用$H(f_i)$和$H(f_j)$的算术和几何平均数。因为$H(f_i)=I(f_i,f_i)$，且是在 Hilbert 空间中对数据进行标准化的，所以一般更倾向于采用几何平均值方法来对数据进行标准化。因此，NMI 可表示如下：

$$NMI(f_i,f_j) = \frac{I(f_i,f_j)}{\sqrt{H(f_i)H(f_j)}} \tag{4-1}$$

根据公式(4-1)我们就可以计算出两个数据属性之间的互信息大小。在得到数据集两两属性之间的标准化互信息值后，我们就可以构建数据集的特征矩阵，该矩阵为一个对称阵，主对角线上的值是属性自身的标准化互信息即$NMI(f_i,f_i)$为 1，其余的为两两属性间的标准化互信息值即$NMI(f_i,f_j)$。特征矩阵$G(\chi)$如图 4-1 所示。

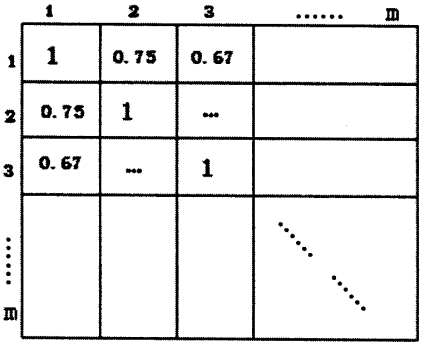


图 4-1 特征矩阵$G(\chi)$

得到特征矩阵G后，可以利用网格划分法将特征矩阵划分为不同的子块，通过搜索子块中互信息的最大值得到数据集中两个属性变量之间的最大信息系数 MIC(χ)，其定义如下：

$$MIC(\chi) = \max_{S<\varepsilon}\{G(\chi)_{i,j}\} \tag{4-2}$$

其中，S为i,j两个变量离散化后所构成的网格面积，它的值要小于阈值ε，ε限制着

实验中我们搜索网格的大小，其值需要在实验中由实验人员设定，ε 过大或者过小都会对实验结果产生比较大的影响，因此选择一个合适的 ε 值是非常重要。

最大信息系数体现的是两个变量之间关联性的大小，关联性越大，属性之间的间隔越小，关联性越小，属性之间的间隔越大。所以在求得数据属性之间的最大信息系数后，我们就可以根据最大信息系数利用最大间隔原理来对数据集进行子空间划分。

数据集有 M 个数据属性，所以在计算数据两两属性间的互信息时，我们可以得到 MMSC 算法的频度为 $f(M)=(M^2-M)/2$，则 MMSC 算法的时间复杂度 $T(M)=o(M^2)$。随着维数 M 的不断增大，算法的时间复杂度不改变，但是算法的执行时间会增加，从而导致算法的执行效率越低。

4.1.2 最大间隔子空间划分

在本文中,我们通过计算数据属性之间的最大信息系数来对子空间的间隔进行判断。根据属性之间相关性的大小来判断变量之间的间隔。MIC 值越大，说明属性之间的相关性越大，反之属性之间的距离越小；MIC 值越小，属性之间的相关性越小，属性之间的距离也越大。为了寻找最大间隔的子空间，$f_h^{(a)}$ 是属于第 h 类中的第 a 个属性，$f_h^{(b)}$ 是属于第 h 类中的第 b 个属性。就需要每一个子空间内各属性的相关性最大所以本文的优化问题可以表示如下:

$$F = Min\sum_{a\neq b} MIC(f_h^{(a)}, f_h^{(b)}) \tag{4-3}$$

其中，MIC 可以用公式(4-2)来进行计算，再结合公式(4-3)和需要划分的子空间数目来对数据属性进行子空间划分。当算法前后两次迭代过程中划分的同一子空间内不同属性之间的 MIC 值之和相差小于我们设定的阈值时，算法就停止迭代，输出子空间划分结果。

综合以上章节所述信息，基于属性最大间隔法划分数据子空间的流程如图 4-2 所示。

4.2 核心算法流程

4.2.1 MMSC 算法子空间划分流程

MMSC 算法的核心部分是对实验数据集进行子空间划分，数据子空间划分的好坏直接影响着最终聚类结果的优劣。从图 4-2 中可知，利用 MMSC 算法对实验数据集进行子空间划分时需要经历四个主要的步骤：计算标准互信息值、构建特征矩阵、搜索最大信息系数、划分数据子空间，因此利用 MMSC 算法划分数据子空间的过程如算法 4.1 所示。

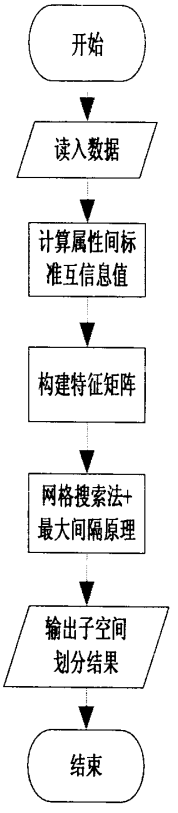


图 4-2 MMSC 算法划分子空间流程图

算法 4.1.

输入：数据集 $\{ x_{\cdot,1}, x_{\cdot,2}, \ldots, x_{\cdot,m} \}$，子空间划分数目 K；

输出：子空间划分结果 C=$\{C_1, C_2, \ldots C_K\}$。

For i=1: m-1

For j=i+1: m

1. 计算属性 i 和属性 j 之间的互信息值，并根据公式(4-1)对互信息值进行归一化处理。
2. 根据第一步得到归一化互信息值构建一个特征矩阵。
3. 根据公式 (4-2)得到数据属性间的最大信息系数。
4. 根据公式 (4-3)将数据划分为 K 个子空间。

End

End

4.2.2 基聚类算法流程

算法 1 是根据最大间隔原理来划分子空间。子空间划分完成后，根据聚类集成原理，我们对数据集进行聚类操作，在聚类过程中，主要使用最大似然估计，步骤描述如下。

算法 4.2.

输入：子空间 C=$\{C_1, C_2, \ldots C_K\}$，聚类数目 K；

输出：子空间聚类集成的正确率 θ。

1. 采用 EM 对划分出来的子空间进行聚类操作，得到不同的基聚类器。

$$\ell(\theta)=\sum_{i=1}^{n}\log p(x;\theta)=\sum_{i=1}^{n}\log\sum_{z}p(x,z;\theta).$$

Repeat

(E step) $Q_i(z^{(i)}):=p(z^{(i)}\left|x^{(i)};\theta\right).$

(M step) $\theta:=\arg\max_{\theta}\sum_{i}\sum_{z^{(i)}}Q_i(z^{(i)})\log\frac{p(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})}.$

Until $\ell(\theta^{(t)})\le\ell(\theta^{(t+1)})$. t：重复迭代的步骤。

2. 利用共识函数 Γ 对基聚类器进行集成操作。
3. 计算子空间聚类集成的准确率。

结合算法 1，2 的步骤， 基于属性最大间隔的子空间聚类集成的工作原理图如图 4-3 所示。

图 4-3 展示了基于属性最大间隔的子空间聚类集成原理图。该算法中关键步骤是对数据集进行子空间划分，在算法开始时，输入参数 k，产生 k 个不同的子空间。此实验中子空间的数目是人为设定的，通过设定不同数目的子空间，可以对比同一数据在划分为不同数目的子空间时，其最终的子空间聚类效果的优劣，可依据得到的实验结果修正划分子空间数目。

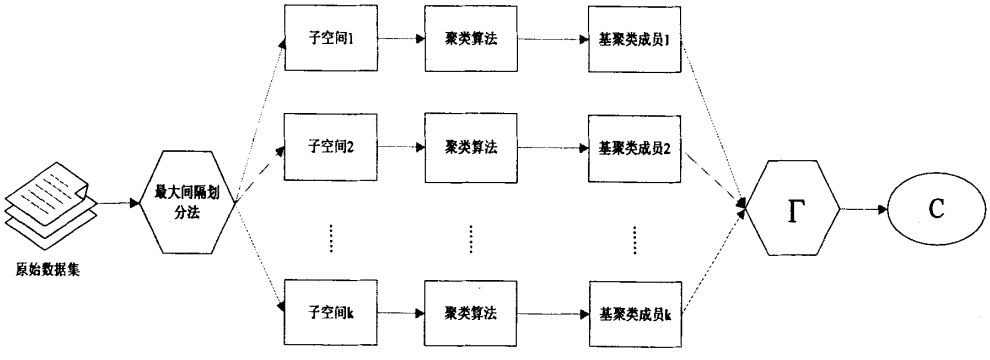


图 4-3 MMSC 算法聚类集成的工作原理图

4.3 本章小结

本章主要介绍了基于属性最大间隔法（MMSC）的子空间聚类。本章首先介绍了MMSC 算法中两个重要的知识点：属性间最大信息系数（MIC）和最大间隔划分法。MIC 算法是通过计算一个数据集中两两变量之间的互信息，然后找出每两个变量之间互信息最大的值通过归一化后构成一个特征矩阵，然后利用网格划分法将特征矩阵划分为不同的子块，通过搜索子块中互信息的最大值得到数据集中两个属性变量之间的最大信息系数 MIC。得到 MIC 值后，再利用最大间隔原理构造 MMSC 算法的目标函数，让数据子空间的划分满足目标函数即可。介绍完 MMSC 算法的理论知识后本章对算法的过程进行了详细的描述，并在这一小节最后总结出了基于属性最大间隔的子空间聚类集成的工作原理。

第 5 章 实验结果与分析

5.1 实验数据集

本次论文有两部分实验，分别是 MRSF 实验和 MMSC 实验。第一部分的 MRSF 实验采用数据集是由 UCI（University of California, Irvine）提供的 10 个机器学习数据集[70]，第二部分的 MMSC 实验采用的是由 UCI 和 NIPS2013 比赛所提供的 12 个数据集，其中 UCI 数据集主要用于检验算法对低维数据集的兼容性，NIPS2013 数据集主要用于验证算法的有效性。MRSF 与 MMSC 实验数据集的相关描述分别如表 5-1 和 5-2 所示。

表 5-1 MRSF 实验数据集的样本,属性和类别描述

数据集	数据样本点数量	属性	类别数
wine	178	13	3
iris	150	4	3
Ionosphere	351	34	2
pima	768	8	2
wdbc	569	30	2
hepatitis	155	19	2
glass	214	9	6
segmentation	2100	19	7
haberman	306	3	2
balance	625	4	3

表 5-2 MMSC 实验数据集的样本,属性和类别描述

数据集	数据样本点数目	属性	类别数
wdbc	569	30	2
Kdd9	1280	41	3
Hepatitis	155	19	2
Krvs	3196	36	2
Ionosphere	351	33	2
Sonar	208	60	2
ad11	3279	1588	2
Secom	1567	590	2
Arcane	100	10000	2
Madelon	2000	500	2
GLI-85	85	22283	2
Hvwnt	606	100	2

5.2 实验评价标准

在聚类研究中有很多评价聚类结果的标准，在本次论文中则采用的是Micro-precision[71]评价标准，即准确率评价标准。准确率评价标准可以直观地评价聚类结果的优劣，因为在利用准确率评价标准对聚类结果进行衡量时需要知道实验数据集原始的分类结果，因此准确率评价标准是属于外部评价法。准确率评价标准的计算公式定义为：

$$MP=\frac{1}{N}\sum_{j=1}^{k}a_j \tag{5-1}$$

其中，a_j表示对数据集某一分类正确的数量，N表示数据集中数据样本点的数量，k表示数据集合中分类的数量。在实验过程中，为了避免偶然性因素对实验结果造成干扰，本次论文中采用取平均准确率来衡量聚类结果的准确性，其计算公式定义为：

$$AMP=\frac{1}{T\times N}\sum_{i=1}^{T}\sum_{j=1}^{k}a_{ij} \tag{5-2}$$

其中，T为重复实验次数，在本次论文实验过程中T的取值为 10 次。

5.3 实验平台介绍

在本次论文进行 MMSC 算法与其他子空间聚类算法的对比实验中，除了 MMSC 算法是在 Matlab 平台上运行的以为，其余算法都是在 WEKA 平台上运行的。Matlab 是由美国 MathWorks 公司研发的商业数学软件，主要用于算法的开发、数据可视化、数据分析和数值计算的高级计算语言和交互式环境。Matlab 软件在算法研究中应用比较广泛，很多研究人员都对此软件都比较熟悉，因此本节重点介绍 WEKA 实验平台。

WEKA 全名为怀卡托智能分析环境(Waikato Environmerit for Knowledge Analysis)。WEKA 是一种利用 JAVA 语言开发的机器学习和数据挖掘工具，在经过 17 年的不断完善发展以后，WEKA 已经成为一款被公认的最成熟的数据挖掘工具之一。与其他操作复杂的现有的数据挖掘工具在进行数据挖掘操作时，大多需要进行复杂的操作，但是 WEKA 却可以让数据挖掘在无需编写代码的情况下实现，使用 WEKA 进行数据挖掘，只需要导入数据，然后选择挖掘算法，调整实验参数即可完成数据挖掘操作。利用 WEKA 进行数据挖掘操作时，WEKA 为实验人员提供了统一的用户操作界面，同时还有数据可视化功能，使得实验人员能够很容易观察数据集的结构。

5.3.1 WEKA 实验平台操作界面

在运行 WEKA 软件的时一共可以选择 4 种不同的用户界面，它们分别是：

Explorer、Experimenter、KnowledgeFlow、SimpleCLI。

(1) Explorer：Explorer 界面是被最为广泛使用的用户界面。实验人员可以从不同的路径读取实验数据，比如：ARFF 格式文件、数据库或者网页。数据集在 WEKA 中打开后，它的一些基本特征将会在用户操作界面中显示出来，例如数据集样本数量以及数据集属性数量。在用户操作界面的右下方会给出数据集的一些可视化图。界面上显示的这些信息是对数据集较为直观的分析，如果要探寻数据集的内部关系，就得使用 WEKA 实验平台提供的各种聚类，分类等算法。算法选定，参数设置完成后，点击“开始”按钮执行算法操作，然后等待 WEKA 运行结果。

(2) Experimenter: Experimenter 界面可以对一组数据集同时进行几个算法的分析操作，随后比较不同算法的实验结果，并从中选择出最优的分析结果。在 Experimenter 界面中还能执行把一项任务分割成多个子项来进行并行化计算的操作，这样可以极大减少算法运行时间。

(3) KnowledgeFlow：KnowledgeFlow 界面为用户提供了一个用来处理大型数据集的递增算法，这个算法有效地克服了 Explorer 界面的不足，因为当用户在 Explorer 界面打开数据集时，所有的数据都将被读入内存中，在数据分析任务规模较小时，数据集全部导入可以提高算法运行效率，但是数据分析任务规模增大时，一般配置的计算机就很难满足算法运行需求。在 KnowledgeFlow 界面的工具条中有预处理工具、可视化模块或者数据挖掘算法等部件，拖动这些部件并放置到画布中，数据流就会随着这些部件的组合而产生。当用户希望对大型的数据集进行分批读取和处理时，就需要选择有递增功能的数据挖掘算法或者过滤器来实现以上功能。

(4) SimpleCLI: SimpleCLI 界面能够让用户通过输入文本命令来达到和其他三种用户界面提供的全部功能。

在 WEKA 平台上运行的子空间聚类算法采用的用户界面是前面介绍的 4 中界面中的 Explorer 界面。WEKA 平台上运行的是 ARFF 格式的数据，下面介绍 ARFF 格式数据。

5.3.2 ARFF 格式数据集

WEKA 平台存储的数据集格式是 ARFF（Attribute-Relation File Format）格式文件，是一种 ASCII 文本文件。ARFF 格式的数据集除去以“%”开始的注释后，整个 ARFF 文件可以分为两个部分：第一是头信息（Head information）部分，头信息中包含了对数据集中数据样本之间的关系和属性的声明；第二是数据信息（Data information）部分，从“@data”标记开始，从那以后的部分即为数据集中给出的数

据信息。在头信息中的关系声明和属性声明描述如下。

关系声明：数据样本间的关系名称定义在 ARFF 格式文件的第一个有效行，其格式为@relation<relation-name>，其中<relation-name>是一字符串。如果在关系声明字符串中含有空格，那么这个字符串就需要加上引号。

属性声明：数据样本的属性声明则是由一列以“@attribute”开始的语句来表示的。每一个“@attribute”语句都是用来定义数据集中对应的一个属性的属性名称和数据类型。在属性声明中，声明语句的顺序很重要，因为它代表了属性在数据集中的位置关系。声明语句中最后声明的属性是 class 属性，即聚类算法对数据集进行聚类的目标任务，class 属性是一个默认的目标变量。ARFF 格式文件中的属性声明的格式为@attribute<attribute-name><datatype>，其中<attribute-name>是必须以字母开头的字符串。属性声明中和前面介绍的关系名称一样，如果属性声明字符串中含有空格，那么这个字符串就需要加上引号。

ARFF 格式的数据样本如图 5-1 所示。

```
% ARFF file for the weather data with some numric features
@relation weather
@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no
```

图 5-1 ARFF 格式数据样本

在进行子空间研究的前期实验阶段，所有的实验数据集都是在 Matlab 平台上进

行的，因此前期搜集的实验数据集都是 Matlab 中的 MAT 格式数据集。WEKA 平台中默认的数据格式是 ARFF 格式，因此在 WEKA 平台上进行 MMSC 算法的对比实验时，需要将 MAT 格式的数据转换为 ARFF 格式的数据。MAT 数据转换为 ARFF 数据步骤如下所示：

(1) 读取.mat 数据集：

```
load filename.mat
```

(2) 生成并打开一个后缀名为.arff 的新文件：

```
fopen('filename.arff','a')
```

(3) 在新文件开头进行关系声明：

```
fprintf(fid,'@RELATION filename\r\n')
```

(4) 对文件进行属性声明：

```
fprintf(fid,'@ATTRIBUTE ')
fprintf(fid,'%5g\t',j)
fprintf(fid,'real\r\n')
```

(5) 在文件中写入数据：

```
fprintf(fid,'@Data \r\n')
```

(6) 关闭.arff 文件：

```
fclose(fid)
```

当数据写入并关闭.arff文件后，由MAT格式数据到ARFF格式数据就转换完成，最终生成的.arff 数据就可以直接导入WEKA平台进行子空间聚类操作。在数据转换过程中，因为我们不知道原有的.mat数据集中每一个数据属性的确切意义，所以在进行数据属性描述的时候我们用数字代表属性意义。图5-2、图5-3分别表示了MAT格式的数据结构和转换为ARFF格式的数据结构。

```
29.62   27.95   27.52   28.94   28.8
59.69   54.68   50.37   57.74   49.19
54.59   52.12   54.98   51.63   47.78
56.99   56.17   51.92   57.16   56
3088.59 3051.03 3056.35 3232.78 3047.33
4112.15 4782.99 4487.42 4312.91 4050.89
409.53  393.75  388.72  388.45  413.1
10956.5 11956.6 11751   12132.9 12057.5
746.4   750.88  718.18  719.88  728.28
58.07   57.97   59.99   59.43   58.04
```

图 5-2 转换前的 MAT 格式数据结构

```
@RELATION hvwnt
@ATTRIBUTE      1     real
@ATTRIBUTE      2     real
@ATTRIBUTE      3     real
@ATTRIBUTE      4     real
@ATTRIBUTE      5     real
@ATTRIBUTE      6     real
@ATTRIBUTE      7     real
@ATTRIBUTE      8     real
@ATTRIBUTE      9     real
@ATTRIBUTE     10     real
@ATTRIBUTE class {1,2}
@Data
29.62    27.95    27.52    28.94    28.8     2
59.69    54.68    50.37    57.74    49.19    1
54.59    52.12    54.98    51.63    47.78    1
56.99    56.17    51.92    57.16    56       2
3088.59 3051.03 3056.35 3232.78 3047.33 2
4112.15 4782.99 4487.42 4312.91 4050.89 1
409.53   393.75   388.72   388.45   413.1    1
10956.5 11956.6 11751    12132.9 12057.5 1
746.4    750.88   718.18   719.88   728.28   1
58.07    57.97    59.99    59.43    58.04    2
```

图 5-3 转换后的 ARFF 格式数据结构

5.4 实验结果分析

5.4.1 基于最小冗余特征子集聚类集成实验结果分析

本节实验对比分析了通过三种方法产生基聚类器后再通过共识函数得到聚类集成结果的聚类准确率，这三种方法分别是：（1）利用 MRFS 方法进行子空间划分后，再通过 K-means 算法对划分的子空间进行基聚类操作；（2）利用 K-means 算法直接对原始数据集进行 10 次基聚类操作；（3）利用 EPDR（Ensembles of Partitions via Data Resembling）[72]算法直接对原始数据集进行 10 次基聚类操作。利用三种不同方法得到基聚类器以后，我们再选用三种不同的共识函数（CSPA，HGPA，MCLA）分别对基聚类器进行集成操作，得到最终的聚类集成结果。最后利用准确率函数对得到的聚类集成结果进行准确率评价，三种共识函数分别对应的准确率结果 CSPA:$\theta_1,\alpha_1,\delta_1$；HGPA:$\theta_2,\alpha_2,\delta_2$；MCLA:$\theta_3,\alpha_3,\delta_3$。实验结果如表 5-3 所示。

从表 5-3 中可以发现利用 MRFS 算法对数据集进行子空间划分后得到的最终聚类集成结果明显优于直接用 K-means 算法和 EPDR 算法得到的聚类集成结果的准确率。

表 5-3 基于 MRFS、K-means、EPDR 三种方法得到的聚类集成准确率对比

Dataset	MRFS			K-means			EPDR		
	θ_1	θ_2	θ_3	α_1	α_2	α_3	δ_1	δ_2	δ_3
wine	**0.8876**	0.5056	0.8708	0.6742	0.4719	0.7022	0.6348	0.5225	0.6901
iris	**0.9667**	0.6400	0.9400	0.8867	0.5400	0.8933	0.8600	0.5267	0.8733
ionosphere	0.7066	0.5385	0.6467	0.6781	0.5840	**0.7123**	0.7009	0.5840	0.7066
pima	0.6914	0.5260	**0.7227**	0.5443	0.5026	0.6602	0.5755	0.5026	0.6615
wdbc	0.8278	0.5132	**0.8840**	0.6696	0.5149	0.8541	0.7188	0.5149	0.8471
hepatitis	0.5742	0.5355	0.5419	**0.6452**	0.5161	0.5935	0.6323	0.5032	0.5871
glass	**0.4720**	0.3925	0.3318	0.3738	0.3271	0.4393	0.4019	0.3738	0.3037
segmentation	**0.5857**	0.3462	0.5262	0.5029	0.4738	0.5433	0.5357	0.4333	0.5305
haberman	0.5098	0.5294	**0.6340**	0.5163	0.5163	0.5229	0.5041	0.5451	0.5328
balance	0.3600	0.4016	0.4896	0.4992	0.4064	0.4112	0.4704	0.4176	**0.5888**

其中，利用 MRFS 算法得到聚类集成结果有 7 次比 K-means 算法和 EPDR 算法的更好，只有 3 次的结果比后面两种算法得到的结果差，因此我们可以从中得到结论，利用 MRFS 算法对原始数据集进行子空间划分后，最终得到的聚类集成结果要优于其余两种算法。从表 5-3 中我们还可以发现在利用 CSPA 作为共识函数时，数据集最终的聚类集成结果准确率一共有 7 次比利用 HGPA 和 MCLA 作为共识函数所得到的聚类集成准确率要高，所以在基于最小冗余特征子集方法对数据集进行子空间划分时，利用 CSPA 作为共识函数所得到的聚类集成结果更加理想。

为了更加直观的观察出 MRFS 算法对 K-means 算法与 EPDR 算法的优越性，本次实验从表 5-3 中总结出了 MRFS 算法对另外两种算法的优势比，分别是 MRFS&K-means 和 MRFS&EPDR。实验对比结果如表 5-4 所示。

表 5-4 MRFS 算法对 K-means 算法与 EPDR 算法的优势比

	θ_1	θ_2	θ_3
α_1	7:3	0	0
α_2	0	6:4	0
α_3	0	0	6:4
δ_1	8:2	0	0
δ_2	0	4:6	0
δ_3	0	0	6:4

从表 5-4 中我们可以直观的发现 MRFS 算法对另外两种算法的优越性，其中除了 MRFS 算法在利用 HGPA 作为共识函数时得到的聚类集成准确率没有 EPDR 算法利用 HGPA 作为共识函数计算出来的准确率好之外，其他情况下，无论算法采用哪一种共

识函数，由 MRFS 算法得到的最终聚类集成结果都要优于其余两种算法得到的聚类集成结果。

通过 MRFS，K-means，EPDR 三种方法得到基聚类器后，为了检验不同方法产生基聚类器的差异性能力，本次实验引入了互信息这一衡量标准。各数据集经不同方法得到的基聚类器之间的互信息如表 5-5 所示。

表 5-5 MRFS、K-means、EPDR 三种方法产生基聚类器之间的互信息值

Dataset	MRFS	K-means	EPDR
wine	**0.2715**	0.4249	0.3406
iris	**0.5284**	0.7412	0.5328
Ionosphere	0.0573	0.1349	0.1123
pima	0.0389	0.0297	0.0300
wdbc	**0.2249**	0.4672	0.4377
hepatitis	**0.0137**	0.0203	0.0174
glass	**0.2158**	0.3918	0.3261
segmentation	**0.3643**	0.5378	0.4329
haberman	0.0190	0.0007	0.0059
balance	**0.0800**	0.1401	0.0848

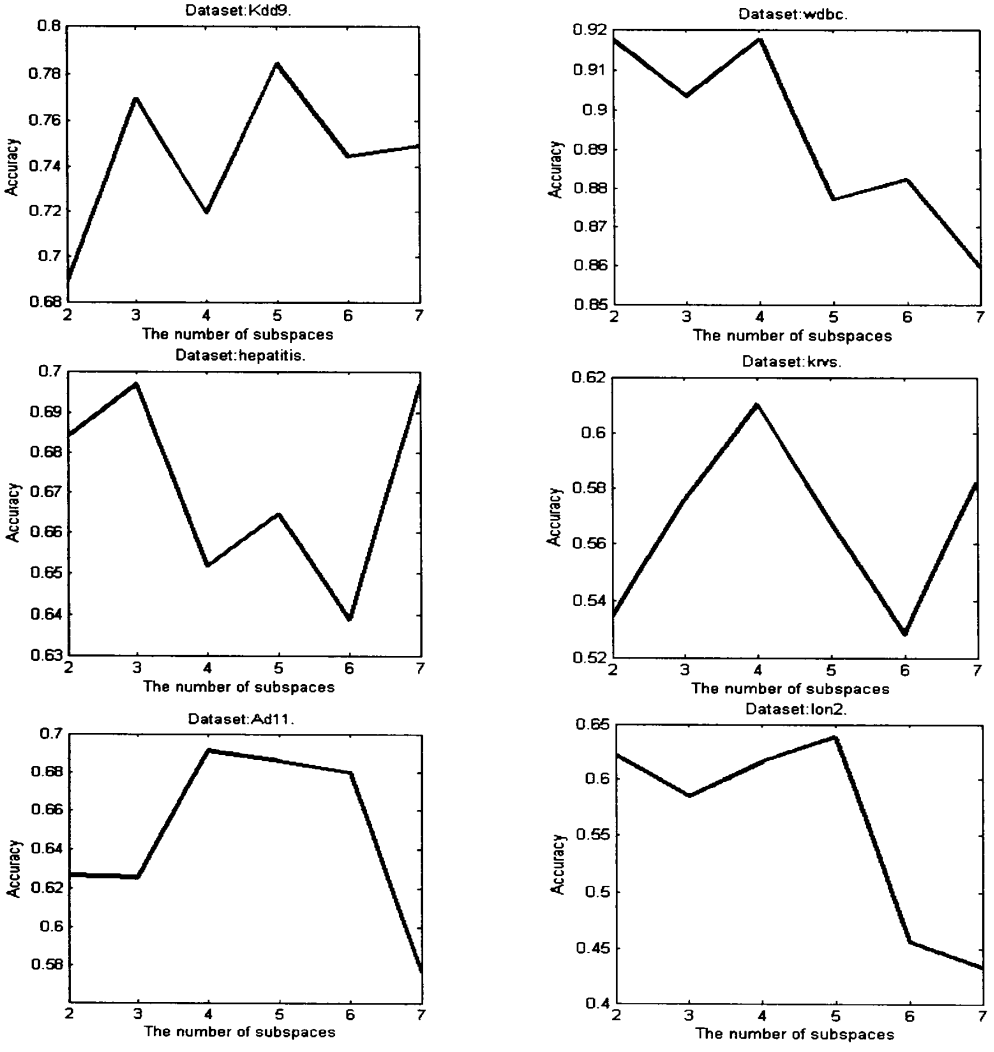
从互信息表 5-5 中我们可以直观地观察出，实验中对数据集进行最小冗余特征子集划分产生数据子空间后，这些数据子集再经过聚类产生的基聚类器之间互信息的值在 10 个数据集中有 7 个数据集比其他两种聚类算法使用没有经过处理的数据集产生的基聚类成员间的互信息值更小，这说明数据集经过子空间划分以后，各个子空间之间的冗余度得到了极大的降低，这样利用产生多样性的基聚类成员，从而提高了数据聚类集成结果的准确率。

5.4.2 基于属性最大间隔的子空间聚类实验结果分析

在子空间聚类实验过程中，数据子空间的划分数目以及共识函数的选取是两个很难把握的因素，因为划分子空间数目的多少以及共识函数选取的合适与否，对最终的聚类结果有着直接的影响。本节在做子空间聚类对比实验以前，会先进行子空间数目的划分以及共识函数的选取进行实验验证，以达到划分出最合适的子空间数目以及选取出最有效的共识函数来做子空间聚类对比实验。

数据子空间的数目对子空间聚类的结果有着较大的影响，数据子空间划分数目这类实验是为了总结划分子空间数目的规律。因为在进行子空间划分时，划分子空间的数目需要设定，但在现实生活中的一些数据集，往往不能确切地知道将数据集划分多少个子空间才能得到比较好的聚类结果。这一类实验第一步是确定划分子空间的数

目，本实验中将每一个数据都划分为 2----7 个子空间，然后利用最大间隔原理对数据集进行子空间划分。实验第二步是采用的是EM算法对划分好的子空间进行聚类操作，得到基聚类结果。在这一步中，如果第一步划分的子空间数目为 2 个，那么经过聚类就得到 2 个基聚类结果，如果子空间数目为 3 个，就得到 3 个基聚类结果，以此类推到划分为 7 个子空间数目时，就得到 7 个基聚类结果。实验第三步是采用 MCLA 作为聚类集成算法，用第二步得到的基聚类结果作为 MCLA 算法的输入数据集。最终的实验结果如图 5-4 所示。从这 12 个标准数据集来看，有八个数据集在子空间数目划分为 4 个时，数据集的聚类集成结果准确率最大，有一个数据集在划分为 2 个子空间时聚类集成结果准确率最大，有一个数据集在划分为 3 个子空间时聚类集成结果准确率最大，有两个数据集在划分为 5 个子空间时聚类集成结果准确率最大。



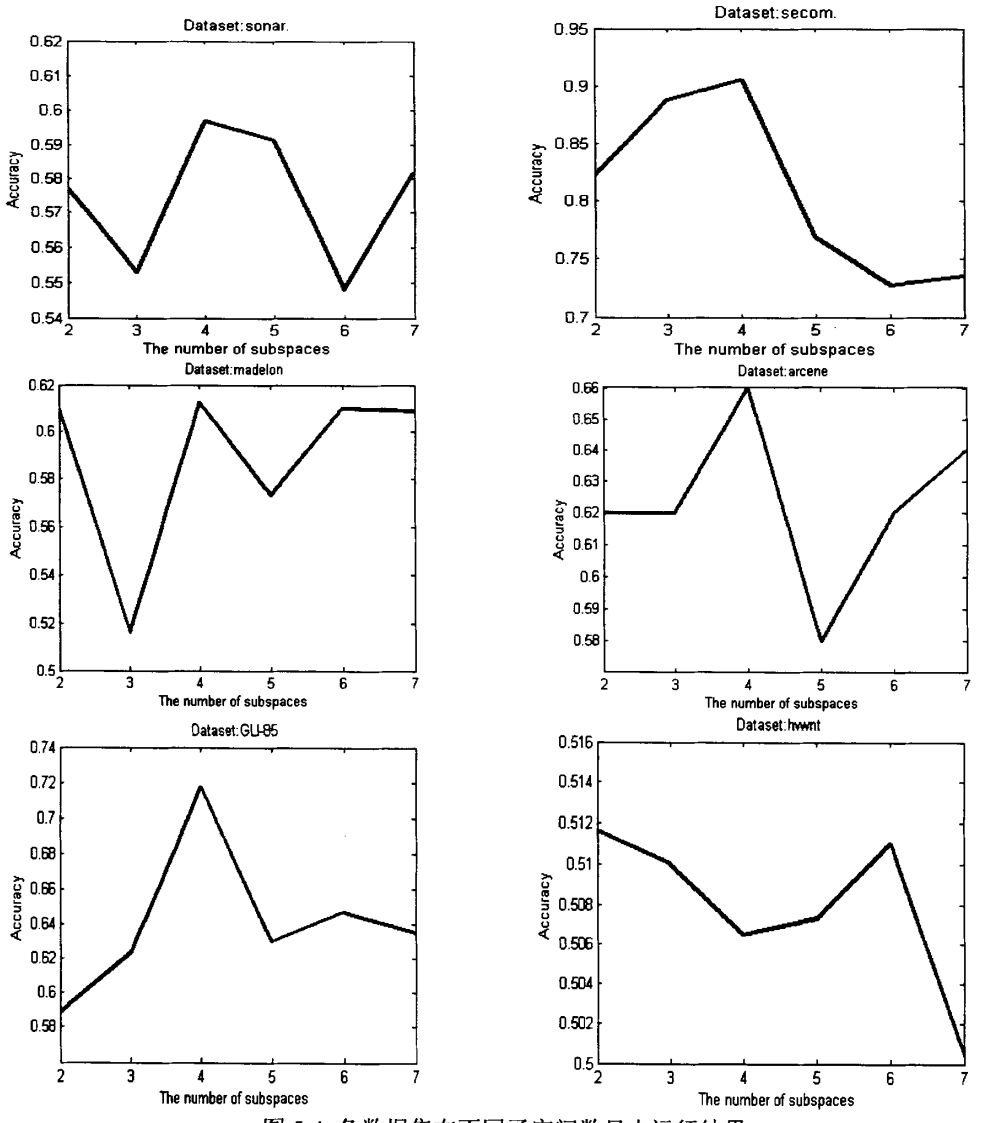


图 5-4 各数据集在不同子空间数目上运行结果

为了更加直观的表现出不同子空间数目对最终结果的影响，我们将实验中 12 标准数据集在划分为相同子空间数目时得到的聚类集成准确率分别相加，得到结果如图 5-5 所示。

从图 5-5 我们可以清晰地看出在子空间数目划分为 4 个的时候，在进行子空间聚类时得到的最终结果比划分其他数目的子空间的结果更好。从图中我们还可以看出对实验数据集进行子空间划分时，划分的子空间数目不是越多越好，在一定范围内，子空间聚类集成的准确率会随着划分的子空间数目增加而提升，但是当划分的子空间数目达到一定的数量后，子空间聚类集成的准确率会随着划分的子空间数目增加而降低。

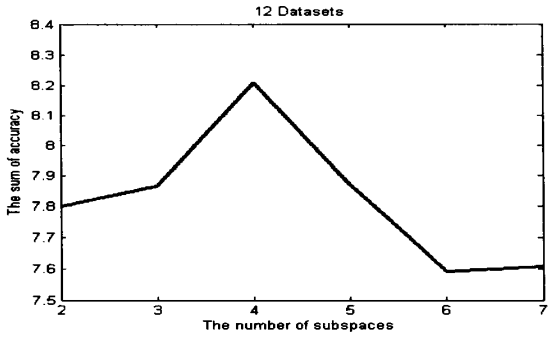


图 5-5 不同子空间数目结果对比图

前面研究了数据集在划分为 4 个子空间时，在大多情况下能够得到最好的聚类集成结果，接下来的实验是共识函数的选取实验。共识函数的选取对最终的聚类集成结果有着十分重要的影响，因此在本节实验中分别选取了 EM ，CSPA，HGPA，MCLA 四种共识函数来对基聚类器进行集成操作。得到的结果如表 5-6 所示。

表 5-6 MMSC 算法在不同共识函数作用下得到的聚类集成结果

Datasets	EM	CSPA	HGPA	MCLA
wdbc	0.85	0.67	0.51	**0.92**
Kdd9	0.66	0.63	0.70	**0.81**
Hepatitis	0.59	0.67	0.54	**0.68**
Krvs	0.52	0.53	0.50	**0.61**
Ionosphere	**0.71**	0.68	0.58	0.62
Sonar	0.55	0.55	0.50	**0.59**
ad11	**0.90**	0.58	0.50	0.89
Secom	0.76	0.50	0.50	**0.91**
Arcane	0.55	0.55	0.50	**0.68**
Madelon	0.57	0.58	0.50	**0.61**
GLI-85	0.68	0.63	0.65	**0.72**
Hvwnt	**0.61**	0.57	0.59	0.51

从表 6 中我们可以看出在采用 MMSC 算法的基础上，将 MCLA 函数作为共识函数时得到的最终聚类集成结果有 9 次比其他三个函数得到的聚类集成准确率更高，只有 3 次的结果比其他函数得到的聚类集成准确率差。表 6 的结果我们可以用图 5-6 更加直观地表示出来。

从图 5-6 中我们可以直观地观察到，在集成过程中采用的基聚类器相同时，选取 MCLA 函数作为共识函数在大多数实验数据集上得到的聚类集成结果的准确率优于采用其他 3 种函数作为共识函数所得到的结果，因此在后面的实验过程中均是采用 MCLA 作为共识函数进行实验。

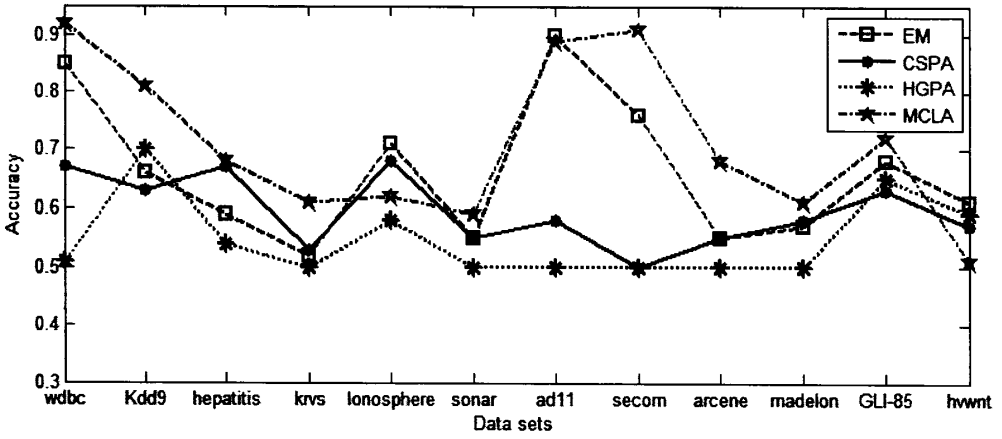


图 5-6 不同共识函数对聚类集成结果影响对比图

本节的前面部分分别进行了数据子空间划分数目的实验以及共识函数选取的实验，接下来的实验是对 MMSC 算法进行对比验证的实验。本次实验选取了 Clique、Proclus、P3c、Fires、MMSC 算法来做实验。Clique、Proclus、P3c、Fires 四种算法是在 WEKA 平台上面运行，MMSC 算法是在 Matlab 平台上面运行。在 WEKA 平台上进行子空间聚类时，最终只能得到单一的正确率值，无法得到数据聚类正确率的标准差，实验中同一数据采用同一种聚类方法进行聚类操作时，我们将算法重复运行 10 次，将 10 次得到的聚类正确率相加然后取平均值记录下来。在 Matlab 平台上运行 MMSC 算法时，我们将同一数据在 MMSC 算法上重复运行 10 次，将 10 次得到的聚类正确率相加然后取平均值记录下来，实验结果记录在表 5-7 中。

在表 5-7 中，用黑体表示该行的最大值，在这 12 个数据集上 MMSC 算法有 9 次获得了最好的聚类正确率；另外 3 个数据分别在 Proclus、P3c、fires 三个算法中取各取得 1 次最大的聚类正确率。所以，表 5-7 总体上体现了以下两个结论：

(1) MMSC 算法在这 12 个数据集上，比 Clique, Proclus, P3c 和 fires 算法在绝大多数情况下都取得了更好的结果。说明在对数据集进行子空间划分的时候，以数据属性间的最大信息系数为依据，利用最大间隔原理来进行子空间划分时数据集中原始的簇类信息丢失较少，使得最终的聚类结果较好。

(2) 总体上，MMSC 算法更适宜各类数据。这是由于 MMSC 模型在对数据集进行子空间划分时，可以对原始数据的离散型和连续型数据都可以进行处理，也可以进行混合处理，让数据在整体上显得更加均衡，使得 MMSC 算法更容易处理各种分布类型的数据。

为了更好地对比这几个算法，我们利用 Friedman Aligned Ranks 检验[73]方法进行了 $1\times n$ 次实验对比。实验结果如表 5-8 所示。本文提出的 MMSC 算法作为基准被对比算法。在表 5-8 的括号中表示的是在 5 个算法以及 12 个数据上进行检验和排序的结

表 5-7 MMSC 算法与其他子空间算法聚类结果

Datasets	Clique	proclus	P3c	fires	MMSC
wdbc	0.63	0.83	0.73	0.63	**0.92**
Kdd9	0.59	0.60	0.57	0.70	**0.81**
hepatitis	0.55	0.63	**0.71**	0.66	0.68
krvs	0.52	0.57	0.52	0.55	**0.61**
Ionosphere	0.64	**0.65**	0.64	0.64	0.62
sonar	0.53	0.54	0.57	0.53	**0.59**
ad11	0.86	0.78	0.84	0.78	**0.89**
secom	0.90	0.90	0.89	0.89	**0.91**
arcene	0.56	0.65	0.57	0.49	**0.68**
madelon	0.50	0.50	0.52	0.55	**0.61**
GLI-85	0.69	0.69	0.69	0.69	**0.72**
hvwnt	0.51	0.47	0.50	**0.53**	0.51

表 5-8 用 Friedman 对 5 个算法的结果进行检验（括号里的排名是用 Friedman Aligned Ranks 检验计算出来且值最小的是最好的结果）

Datasets	Clique	proclus	P3c	fires	MMSC	Total
wdbc	-0.118(59.5)	0.082(4)	-0.018(42)	-0.118(59.5)	0.172(1)	166
Kdd9	-0.064(55)	-0.054(54)	-0.084(56)	0.046(10)	0.156(2)	177
hepatitis	-0.096(57)	-0.016(41)	0.064(6)	0.014(18)	0.034(12)	134
krvs	-0.034(51)	0.010(22)	-0.010(37.5)	-0.010(37.5)	0.050(9)	157
Ionosphere	0.002(28)	0.012(20)	0.002(28)	0.002(28)	-0.018(43)	147
sonar	-0.022(45.5)	-0.012(39)	0.018(16)	-0.022(45.5)	0.038(11)	157
ad11	0.030(13)	-0.050(52.5)	0.010(21)	-0.050(52.5)	0.060(7)	146
secom	0.002(25.5)	0.002(25.5)	-0.008(35.5)	-0.008(35.5)	0.012(19)	141
arcene	-0.030(47)	0.060(8)	-0.020(44)	-0.100(58)	0.090(3)	160
madelon	-0.036(49.5)	-0.036(49.5)	-0.016(40)	0.014(17)	0.074(5)	161
GLI-85	-0.006(32.5)	-0.006(32.5)	-0.006(32.5)	-0.006(32.5)	0.024(15)	145
hvwnt	0.006(23.5)	-0.034(48)	-0.004(30)	0.026(14)	0.006(23.5)	139
Total	487	396	388.5	408	150.5	
Average	40.5833	33	32.375	34	12.5417	

果。 我们按各个算法在不同数据集上所得实验结果的排名之和的平均值大小来体现算法的优劣，MMSC 算法以平均值 12.5417 排在首位；P3c 算法以 32.375 排在第二位；proclus 算法以 33 排在第三位；fires 以 34 排在第四位；Clique 以 40.5833 排在最后一位。Friedman 排名实验是为了检验多种算法在多个数据集上实验所得的结果是否具有显著性差别。

$$\sum_{j=1}^{k} \hat{R}_{\cdot j}^{2} = 487^2 + 396^2 + 388.5^2 + 408^2 + 150.5^2 = 734030,$$

$$\sum_{j=1}^{k} \hat{R}_{i\cdot\cdot}^2 = 166^2 + 177^2 + 134^2 + \cdots + 139^2 = 280812,$$

$$T = \{(5-1)(734030-(5\times12^2/4)(5\times12+1)^2)\} \div$$
$$\{(5\times12(5\times12+1)(2\times5\times12+1))/5-(1/5)\times280812\} = 7.9297 。$$

实验中我们用了5个算法和12个数据进行实验，T 是有 $5-1=4$ 个自由度的卡方分布。根据 $x^2(4)$ 分布，我们计算出P值分别为0.00068454（单尾）和0.00136908（双尾）。我们可以发现这些值远小于0.05，这表明不同算法得到的结果具有很大的显著性差异。

5.5 本章小结

本章主要对第三章和第四章提出的两个不同的子空间划分法进行了实验验证。在对基于最小冗余法进行验证时，主要对 MRFS 算法与其他两个聚类算法进行了聚类准确率，聚类优势对比，这两组对比实验表明采用 MRFS 算法对数据集进行子空间划分后得到的聚类结果均优于其余两个聚类算法，在对三种不同方法产生的基聚类器之间的互信息进行对比时发现数据集经过子空间划分以后，各个子空间之间的冗余度得到了极大的降低，这样利用产生多样性的基聚类成员。在对基于属性最大间隔法进行验证时，主要进行了子空间聚类算法中划分子空间数目的研究，以及集成时采用的共识函数的研究，在确定了子空间划分数目以及所采用的共识函数以后，本论文对 MMSC 算法与其余子空间聚类算法进行了聚类准确率对比实验，实验结果表明采用 MMSC 算法对数据集进行子空间划分后得到的聚类结果比其他子空间聚类算法在相同数据集上面得到的聚类准确率更高。因为 MMSC 算法与其他子空间聚类算法的对比实验是在不同的实验平台上进行的，所以本次对比实验没有进行算法的时间复杂度对比。最后本论文对不同算法的聚类结果进行了 Friedman Aligned Ranks 验证，验证结果表明 MMSC 算法较其他子空间聚类算法有较大优势。

总结与展望

聚类之所以能成为众多研究者研究的热点课题，是因为它在机器学习领域有着极其重要的价值，是无监督学习中的代表。聚类能够发现数据集的内在分布结构。聚类技术利用相似性计算方法判断数据样本之间的相似性，根据不同数据样本之间的相似性大小把数据集分割成不同的类，划分类所遵循的规则是属于不同类的数据样本点之间相似性最小，属于同一类的数据样本点之间相似性最大。随着社会与人类生产实践的发展，产生了很多的大数据，这些大数据具有数据体量巨大，数据类型繁多，价值密度低和处理速度快等特点。其中，数据体量巨大主要在数量和维度上体现，而子空间聚类就是解决高维数据的方法之一，在另外一个方面，传统的许多聚类算法受大数据的“维度效应”影响，其聚类有效性会大大降低。高维数据的处理就成为现阶段机器学习面临的主要任务之一。

本文针对高维数据的子空间划分问题，提出了两种划分方法，第一种是基于最小冗余的子空间划分法，第二种是基于属性最大间隔的子空间划分法。基于最小冗余特征子集的子空间划分法是在 K-means 算法的基础上改进的，将计算数据特征变量间的互信息替换 K-means 算法中计算数据特征变量间的距离，根据数据特征变量间互信息值的大小来对数据进行子空间划分。基于属性最大间隔的子空间划分法是通过计算数据两两属性间的互信息，然后将属性间的互信息值构成一个特征矩阵。得到特征矩阵后，再利用网格划分法将特征矩阵划分成不同的子块，通过搜索子块中互信息的最大值得到数据集中两个属性变量之间的最大信息系数，最大信息系数体现了两个属性之间关联性的大小，关联性越大，属性间间隔越小，关联性越小，属性间间隔越大，因此在得到最大信息系数后，我们就可以利用最大间隔原理来对数据集进行子空间划分。在解决了子空间的划分问题后，本文还对子空间划分的数目，以及进行聚类集成时采用的共识函数进行了探索研究。综上所述，本文的主要工作如下：

(1) 对子空间聚类集成研究的背景和意义进行了论述，并对子空间聚类以及子空间聚类集成的国内外现状进行了总结。

(2) 介绍了有关聚类方面的一些基本概念、研究方向和现阶段已有的研究成果。详细介绍了聚类算法在利用数据样本点间的距离大小对数据样本点进行相似性度量时采用的几种距离公式，然后介绍了现阶段已有的聚类算法分类情况。最后对聚类集成进行了详细的阐述，并针对聚类集成中基聚类成员的产生与共识函数的设计两个关键步骤进行了详细介绍。

(3) 提出了最小冗余法对数据进行子空间的划分。在引入最小冗余法之前介绍了衡量变量间相关性的三种常用相关系数的特点以及如何利用回归直线方程来反应变量间的数量关系。随后重点介绍了如何基于最小冗余对数据特征变量进行子空间的划分，通过计算特征变量之间的互信息值，通过利用变量间最

大相关性，并加入最小冗余来限制特征变量的选择，最终实现对数据子空间的划分。

(4) 提出了基于属性最大间隔法（MMSC）的子空间聚类。MMSC 算法中两个重要的知识点：属性间最大信息系数（MIC）和最大间隔划分法。MIC 算法是通过计算一个数据集中两两变量之间的互信息，然后找出每两个变量之间互信息最大的值通过归一化后构成一个特征矩阵，然后利用网格划分法将特征矩阵划分为不同的子块，通过搜索子块中互信息的最大值得到数据集中两个属性变量之间的最大信息系数 MIC。得到 MIC 值后，再利用最大间隔原理构造 MMSC 算法的目标函数，让数据子空间的划分满足目标函数即可。介绍完 MMSC 算法的理论知识后又介绍了数据挖掘工具 WEKA，具体介绍了 WEKA 的用户操作界面以及本次在 WEKA 平台上采用数据的格式。

(5) 通过对 MRFS 算法和 MMSC 算法进行验证，实验结果表明利用这两种算法进行子空间划分后，对数据进行聚类操作能够得到比其他子空间算法更好的聚类结果。

子空间聚类是近几年聚类领域的研究热点，本文提出的两种子空间划分法在一定程度上提升了聚类算法在大数据上的聚类准确率。虽然本次论文对大数据中子空间的划分数目，以及聚类集成过程共识函数的选取也做了一定的研究，但由于时间比较短暂，很多工作都是在已有的算法基础上进行的，在今后的研究过程中，还需要进一步对共识函数进行深入分析和探讨。同时在子空间划分方法的问题上，也还需要做进一步的研究，以提高算法的广泛性和实用性。

致 谢

回望过去，不由地感叹时光如同白驹过隙，三年的研究生生涯即将接近尾声。在论文完成之际，首先我要衷心的感谢我的导师王红军老师，在西南交通大学三年的研究生时间里，王老师在学习和生活等方面一直给予了我非常多的关怀和帮助，在论文的实验与写作中予以悉心的指导和教诲，并且给我提供了许多参与实验室讨论和学习的机会，每次与王老师的学术交流和讨论都能够让我受益匪浅，王老师认真负责的教学态度和在机器学习和数据挖掘领域的学术造诣，都为我以后的工作学习树立了榜样。

感谢杨燕老师对本次论文完成的关心与指导，在每一次实验室的讨论会上杨燕老师都会关心我的论文进度，对我在论文写作中遇到的困难提供解决的意见，杨燕老师对待学生平易近人、对待学术严谨认真的态度是我永远学习的典范。

感谢云计算与智能技术实验室的所有老师，是您们在实验室为同学们营造了一个轻松愉悦的学术交流氛围，为我们在日常的学习生活中答疑解惑，通过与您们的交流沟通，不仅丰富了我们的科学文化知识，同时也教会了我们应该怎样与人相处，增进了同学们之间的感情，让我们的实验室更加团结，积极，上进。

感谢聚类小组的同学对我学习、生活上的帮助，在此特别感谢陈云风，黄树东，杨琪，肖丽莎，杨静娴等同学对本文提供的意见和帮助，感谢实验室的各位师兄师姐和师弟师妹们对我三年研究生生活和学习的帮助。

感谢我的家人，尤其是我的父母在学习期间给我的精神和物质支持，是您们无怨无悔的付出和默默的关怀，才让我可以全身心的投入到学习研究中，感谢您们的关心与支持。

感谢各位评委老师在百忙之中抽出时间评阅本文以及各位答辩组专家为评审本文所付出的辛勤汗水。

最后，感谢每一位在我成长道路上给我帮助和关心的人！谢谢您们……

参考文献

[1] J Han, M L Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2001.

[2] J M Queen. Some methods for classification and analysis of multivariate observation [C]. In proc. 5th Berkeley Sump. Math. Statist, 1967:281-297.

[3] L Kaufman, P J Rousseuw. Finding groups in data: an introduction to cluster analysis [M]. New York: John Wiley and Sons, 1990.

[4] J C Bezdek. Pattern recognition with fuzzy objective. Function Algorithm [M]. Plenum Press. New York, 1981.

[5] H P Kriegel, P Kroger, A Zimek. Clustering high dimensional data: A survey on subspace clustering, pattern based clustering, and correlation clustering. ACM Transactions on Knowledge Discovery from data, 2009, 3(1):1-58.

[6] E Leopold, J Kindermann. Text categorization with support vector machines: How to represent texts in input space. Machine Learning, 2002, 46(1-3):423-444.

[7] M Verleysen. Learning high-dimensional data. Limitations and Future Trends in Neural Computation. Siena: IOS Press, 2003:141-162.

[8] L Parsons, E Haque, H Liu. Subspace clustering for high dimensional data: A review. ACM SIGKDD Explorations Newsletter, 2004, 6(1):90-105.

[9] G Moiseg, A Zimeka, P Kroger, H P Kriegel, J Sander. Subspace and projected clustering: Experimental evaluation and analysis. Knowledge and Information Systems, 2009, 21(3): 299-326.

[10] C C Aggarwal, C Procopiuc, J L Wolf, P S Yu, J S Park. Fast algorithm for projected clustering. ACM-SIGMOD. New York: ACM Press, 1999:61-71.

[11] 阳琳赟, 王文渊. 聚类融合方法综述. 计算机应用研究, 2005, 12(3):8-12.

[12] J.W Han, M Kamber. 数据挖掘：概念与技术[M]. 北京: 机械工业出版社, 2001.

[13] J Z Huang, M K Ng, H Rong, Z Li. Automated variable weighting in k-means type clustering. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2005, 27(5):657-668.

[14] R Agrawal, J Gehrke, D Gunopulos, P Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In Proceedings of the ACM International Conference on Management of Data (SIGMOD), Seattle, WA, 1998.

[15] C H Cheng, A W Fu. Y Zhang. Entropy-based subspace clustering for mining numerical data[C]. Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM Press, 1999: 84-93.

[16] S Goil, H Nagesh, A Choudhary. MAFIA: Efficient and scalable subspace clustering for very large data sets[C]Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999: 443-452.

[17] H S Nagesh. High performance subspace clustering for massive data sets. Master's thesis, Northwestern University, 2145 Sheridan Road, Evanston IL 60208, June 1999.

[18] H S Nagesh, S Goil, A Choudhary. A scalable parallel subspace clustering algorithm for massive data sets. June 2000.

[19] J W Chang, D S Jin. A new cell-based clustering method for large, high-dimensional data in data mining applications. In Proceedings of the 2002 ACM symposium on applied computing, ACM Press, 2002:503-507.

[20] B Liu, Y Xia, P S Yu. Clustering through decision tree construction. In Proceedings of the ninth international conference on Information and knowledge management. ACM Press, 2000: 20-29.

[21] C M Procopiuc, M Jones, P K Agarwal, T. M. Murali. A Monte Carlo algorithm for fast

projective clustering. In Proceedings of the 2002 ACM SIGMOD international conference on Management of data. ACM Press, 2002:418-427.
[22]C C Aggarwal, J L Wolf, P S Yu, C Procopiuc, J S Park. Fast algorithms for projected clustering. In Proceedings of the 1999 ACM SIGMOD international conference on Management of data. ACM Press, 1999:61-72.
[23]C C Aggarwal, P S Yu. Finding generalized projected clusters in high dimensional spaces. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data. ACM Press, 2000:70-81.
[24]K G Woo, J H Lee. FINDIT: a Fast and Intelligent Subspace Clustering Algorithm using Dimension Voting. PhD thesis, Korea Advanced Institute of Science and Technology, Taejon, Korea, 2002.
[25]J Yang, W Wang, H X Wang, P Yu. δ-clusters: capturing subspace correlation in a large dataset. In Data Engineering, 2002. Proceedings. 18th International Conference, 2002: 517-528.
[26]J H Friedman, Jacqueline [J].Meulman. Clustering objects on subsets of attributes. 2002.
[27]D D Wang, C Ding, T Li. K-Subspace Clustering. European Conference, ECML PKDD 2009. 2009: 506-521.
[28]I Assent., E Muller., SGunnemann, R Krieger, T Seidl. 1st International Workshop on Discovering, Summarizing and Using Multiple Clustering (MultiClust 2010) in conjunction with 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2010). 2010.
[29]E Elhamifar, R Vidal. Sparse subspace clustering[C].Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009: 2790-2797.
[30]M Soltanolkotabi, E Elhamifar, E Candes. Robust subspace clustering[J].arXiv preprint arXiv:1301.2603, 2013.
[31]Y X Wang, H Xu. Noisy sparse subspace clustering[J]. arXiv preprint arXiv:1309.123 3, 2013.
[32]F Gullo, C Domeniconi, A Tagarelli. Projective Clustering Ensembles. Data Mining (ICDM), 2009 IEEE 9th International Conference. 2009:794-799.
[33]F Gullo, C Domeniconi, A Tagarelli. Enhancing Single-Objective Projective Clustering Ensembles. Data Mining (ICDM), 2010 IEEE 10th International Conference. 2010:833-838.
[34]F Gullo, C Domeniconi, A Tagarelli. Advancing Data Clustering via Projective Clustering Ensemble. Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2011.
[35]管仁初. 半监督聚类算法的研究与应用. 吉林大学, 博士学位论文, 2010.
[36]R Xu, D Wunsch. Survey of clustering algorithms. IEEE transactions on neural networks, 2005. 12(3): 645-678
[37]J W Han, M Kamber, 范明(译), 孟小峰(译). 数据挖掘概念与技术, 北京: 机械工业出版社, 2007.
[38]A Strehl, J Ghosh. Cluster ensembles-a knowledge reuse framework for combining multiple partitions [J]. Machine Learning Research. 2002, 3:583-617.
[39]A Topchy, A K Jain, W Punch. A Mixture Model for Clustering Ensembles [C]. Proceedings of the 4th SIAM International Conference on Data Mining, 2004:379-390.
[40]B M Bidgoli, A Topchy, W F Punch. A Comparison of Resampling Methods for Clustering Ensembles [C]. Intl. Conf. on Machine Learning, Models, Technologies and Applications (MLMTA 2004), 2004:939-945.
[41]A L N Fred, A K Jain. Data clustering using evidence accumulation. Proceedings of the 16th International Conference on Pattern Recognition, Quebec ,2001:276-280.

[42]Y Yang, M Kamel. An aggregated clustering approach using multi-ant colonies algorithms. Pattern Recognition, 2006, 39(7): 1278-1289.
[43]罗会兰. 聚类集成关键技术研究. 浙江大学，博士学位论文, 2007.
[44]B M Bidgoli, A Topch, W F Punch. Ensembles of Partitions via Data Resampling [C]. Proceedings International Conference on Information Technology, Coding and Computing (ITCC 2004) , 2004,2:188-192.
[45]S Dudoit, J Fridlyand. Bagging to Improve the Accuracy of a Clustering Procedure [J].Bioinformatics, 2003, 19 (9): 1090-1099.
[46]B Fischer, J M Buhmann. Path-based Clustering for Grouping of Smooth Curves and Texture Segmentation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25 (4): 513-518.
[47]S S Chen, B Tung. The Common Intrusion Detection Framework Architecture [EB /OL].http: / /www.isi.edu/ gost / cidf /drafts/2001 / architecture.txt, 2003.
[48]A Topchy, B M Bidgol, A K Jain, et al. Adaptive Clustering Ensembles [C] proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004) Volume-1,2004:272-275.
[49]L I Kuncheva, S T Hadjitodorov Using Diversity in Cluster Ensembles [C] Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2004: 1214-1219.
[50]A Topchy, A K Jain, W F Punch. Combining Multiple Weak Clusterings [C] Proceedings of the 3 rd IEEE International Conference on Data Mining (ICDMp03), 2003:331-338.
[51]A L Fred. Finding Consistent Clusters in Data Partitions[C]. Proceedings of the 2nd International Workshop on Multiple Classifier Systems, Volume 2096 of Lecture Notes in Computer Science, Sp ringer, 2001:309-318.
[52]A Topchy, A K Jain, Punch W. Clustering ensembles: models of consensus and weak partitions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(12): 1866-1881.
[53]Z Zhou, W Tang. Cluster ensemble. Knowledge-Based Systems. 2006, 19(1): 77-83.
[54]Y Yang, M Kamel, F Jin. ART-based clustering aggregation. Proceedings of the IEEE International Conference on Granular Computing, Atlanta ,2006:482-485.
[55]Y H Luo, S C Xiong. Clustering ensemble for unsupervised feature selection. Proceedings of the 6th International Conference on Fuzzy Systems Knowledge Discovery, Tianjin ,2009:445-448.
[56]李杉, 张化祥. 基于Bagging的聚类集成方法. 计算机工程与设计, 2010:164～166.
[57]C Duan, J C Huang, B Mobasher. A consensus based approach to constrained clustering of software requirements. Proceedng of the 17th ACM Conference on Information and Knowledge Management, Napa Valley ,2008:1073-1082.
[58]A Fred, A K Jain. Data Clustering Using Evidence Accumulation[C]. Proceedings of the 16th International Conference on Pattern Recognition (ICPR 2002), 2002,4:276-280.
[59]A Fred, A K Jain. Evidence Accumulation Clustering Based on the K-means Algorithm [C]. Proceedings of the International Workshop son Structural and Syntactic Pattern Recognition (SSPR 2002), 2002:442-451.
[60]H Ayad, O A Basir, M Kamel. A Probabilistic Model Using Information Theoretic Measures for Cluster Ensembles [C]. Proceedings of the 5th International Workshop on Multiple Classifier Systems, Volume 3077 of Lecture Notes in Computer Science, Sp ringer, 2004:144-153.
[61]H Ayad, M Kamel. Finding Natural Clusters Using Multi Clusterer Combiner Based on sharred Nearest Neighbors[C]. Proceedings of the 4th International Workshop on

Multiple Classifier Systems (MCSp03), Volume 2709 of Lecture Notes in Computer Science, Springer, 2003:166-175.

[62]H Ayad, M Kamel. Refined Shared Nearest Neighbors Graph for Combining Multiple Data Clusterings, Advances in Intelligent Data Analysis [C]. Proceedings of the 5th International Symposium on Intelligent Data Analysis (IDA 2003), Volume 2810 of Lecture Notes in Computer Science, Sp ringer, 2003:307-318.

[63]P E Jouve, N Nicoloyannis. A New Method for Combining Partitions, Applications for Distributed Clustering [C]. International Workshop on Paralell and Distributed Machine Learning and Data Mining (ECML/PK- DD03), 2003.

[64]X Z Fern, C E Brodley. Solving Cluster Ensemble Problems by Bipartite Graph Partitioning [C]. The 21st International Conference on Machine Learning, 2004.

[65]A Webb, Statistical Pattern Recognition. Arnold, 1999.

[66]A K Jain, R P W Duin, J Mao, "Statistical Pattern Recognition: A Review," IEEE Trans. Pattern Analysis and Machine Intelligence, 2000, 22 (1):4-37.

[67]A K Jain, D Zongker, "Feature Selection: Evaluation, Application, and Small Sample Performance," IEEE Trans. Pattern Analysis and Machine Intelligence, 1997, 19(2): 153-158.

[68]C Ding, H C Peng, "Minimum Redundancy Feature Selection from Microarray Gene Expression Data," Proc. Second IEEE, Computational Systems Bioinformatics Conf. 2003:523-528.

[69]D N Reshef, Y A Reshef, H K Finucane. Detecting novel associations in large data sets. Science, 2011,334 (6062): 1518-1524.

[70]Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets.html.

[71]Z H Zhou, W Tang. Cluster ensemble. Knowledge Based Systems, 2006, 19(1): 77-83.

[72]B M Bidgoli, A Topch, W F Punch. Ensembles of Partitions via Data Resembling [C]. Proceedings International Conference on Information Technology, Coding and Computing (ITCC 2004) , 2004,2:188-192.

[73]S Garcla, A Fernndez, J Luengo, F Herrera. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences, 2010. 180: 2044–2064.

攻读硕士学位期间发表的论文

1.发表的论文：

[1] Bo Liu, Hong-Jun Wang, Yan Yang, Xiao-Chun Wang. The Method of Cluster Ensemble Based on Minimum Redundancy Feature Subset , 2012 Second International Conference on Electronics, Communications and Control.2012：2320-2323.

[2] 刘波,王红军,成聪,杨燕.基于属性最大间隔的子空间聚类.南京大学学报(自然科学版),已录用,2014.

2.参与的科研项目：

[1] 国家自然科学基金项目：半监督聚类集成的关键技术研究（项目编号：61003142）. 起止时间:2011.01 - 2013.12.

[2] 国家自然科学基金项目：藏文 WEB 信息的社会网络动态演化机理研究（项目编号: 61262058）.起止时间:2013.01 - 2016.12.

[3] 国家自然科学基金项目: 基于半监督学习的聚类集成机理及高效算法研究（项目编号: 61170111）. 起止时间: 2012.1-2015.12.

[4] 西南交通大学牵引动力国家重点实验室自主研究课题: 基于云计算的海量高铁数据处理关键技术研究（项目编号：2012TPL_T15）. 起止时间：2012.1-2014.12.